SITRAIN
Training for Industry

SIMATIC TIA Portal S7-1500

Cours TIA-PRO1

Nom : Diaby

Cours : du ________________ au ____________________

Formateur : ______________________________________

Lieu : ______________________________________

Offre formation SITRAIN sur Internet :
www.siemens.com/sitrain

Mise à jour: MC (05/03/18)

Cours en version : (pour STEP7 Version 15a)

Sommaire

1

Mr Aboubacar Sidiki DIABY



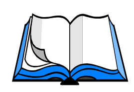
1. Matériel utilisé pour la formation et adressage → Poste de travail équipé d'un S7-1500

À l'issue du chapitre, vous allez...



... connaître la configuration du poste de travail

... savoir câbler les périphéries du poste de travail



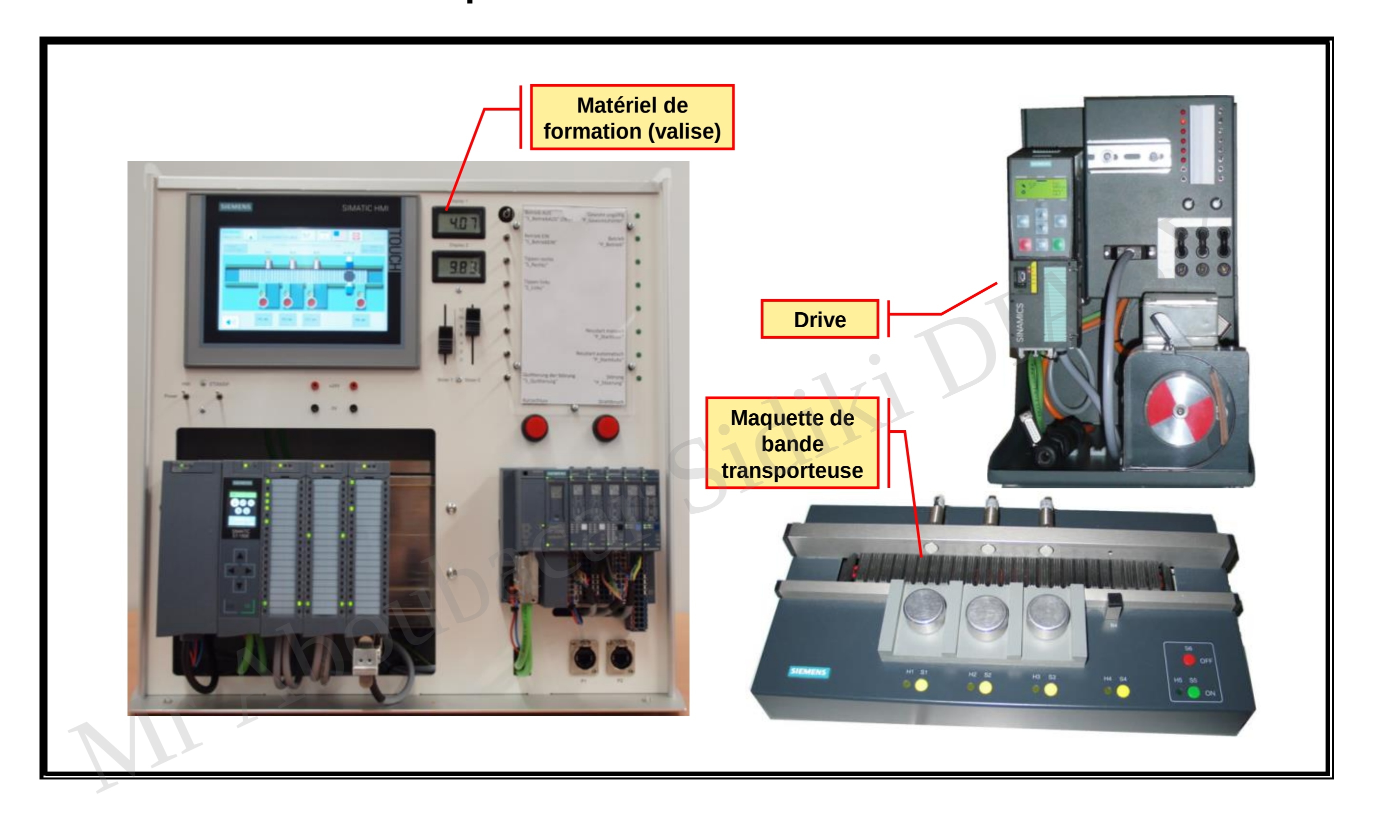
Ce chapitre présente la configuration du poste de travail utilisé pendant le cours.

A l'issue de ce chapitre, vous connaîtrez l'adressage d'une CPU S7-1500. Vous aurez une vue d'ensemble du fonctionnement et du câblage des différentes composantes du poste de travail.

Objectifs pédagogiques :
Savoir câbler et adresser les composantes matérielles du poste de travail.

Mr Aboubacar Sidiki DIABY

1.1. Présentation du poste de travail

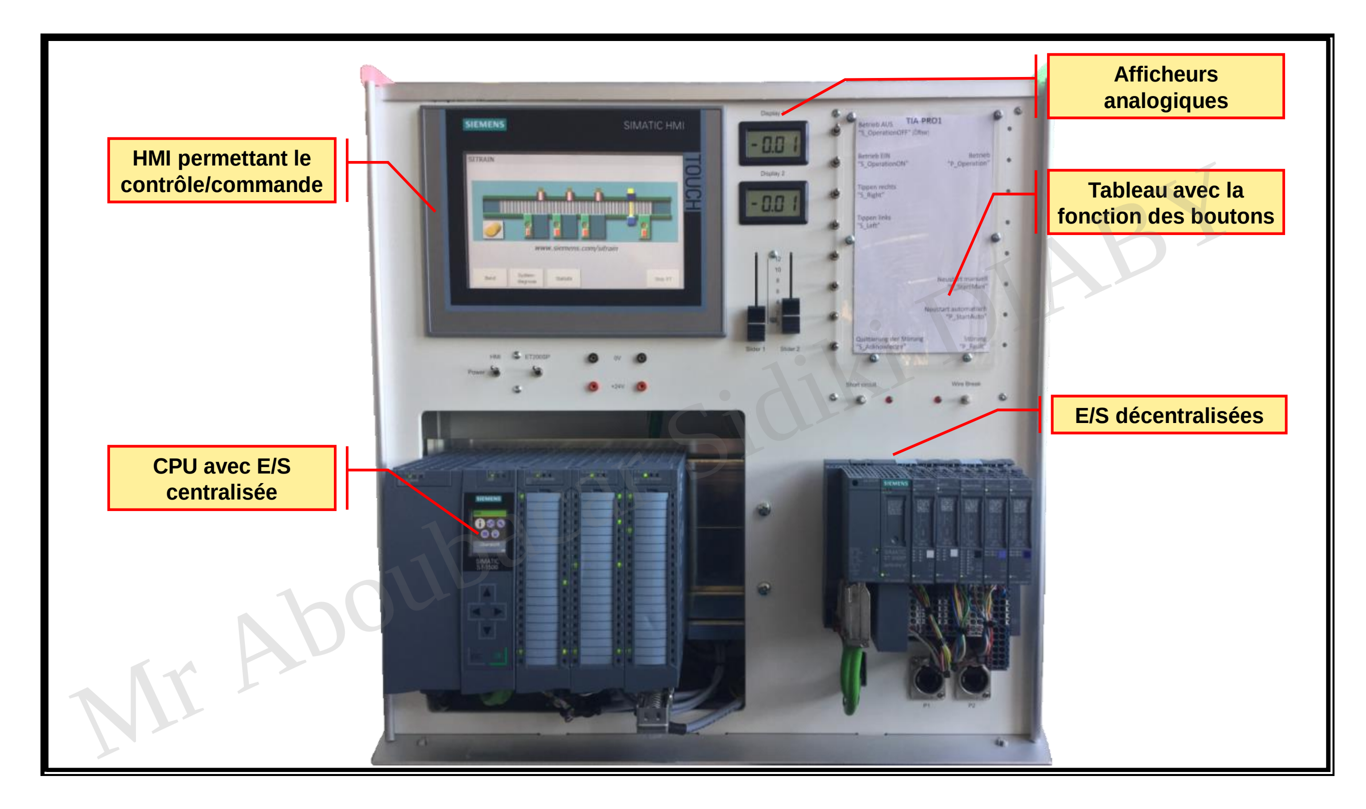


Composantes matérielles du poste de travail S7-1500 utilisé pour la formation

Le poste de travail comprend :

- Une console SIMATIC Field PG
- Une valise contenant le matériel de formation avec un automate S7-1500, une station périphérique ET200SP, un pupitre à écran tactile, des organes de commande (commutateurs/boutons et curseurs de réglage) et des organes d’affichage (LED et afficheurs de tension)
- Un variateur SINAMICS G120
- Une maquette de bande transporteuse (convoyeur à bande)

1.2. Poste de travail équipé d'un automate S7-1500

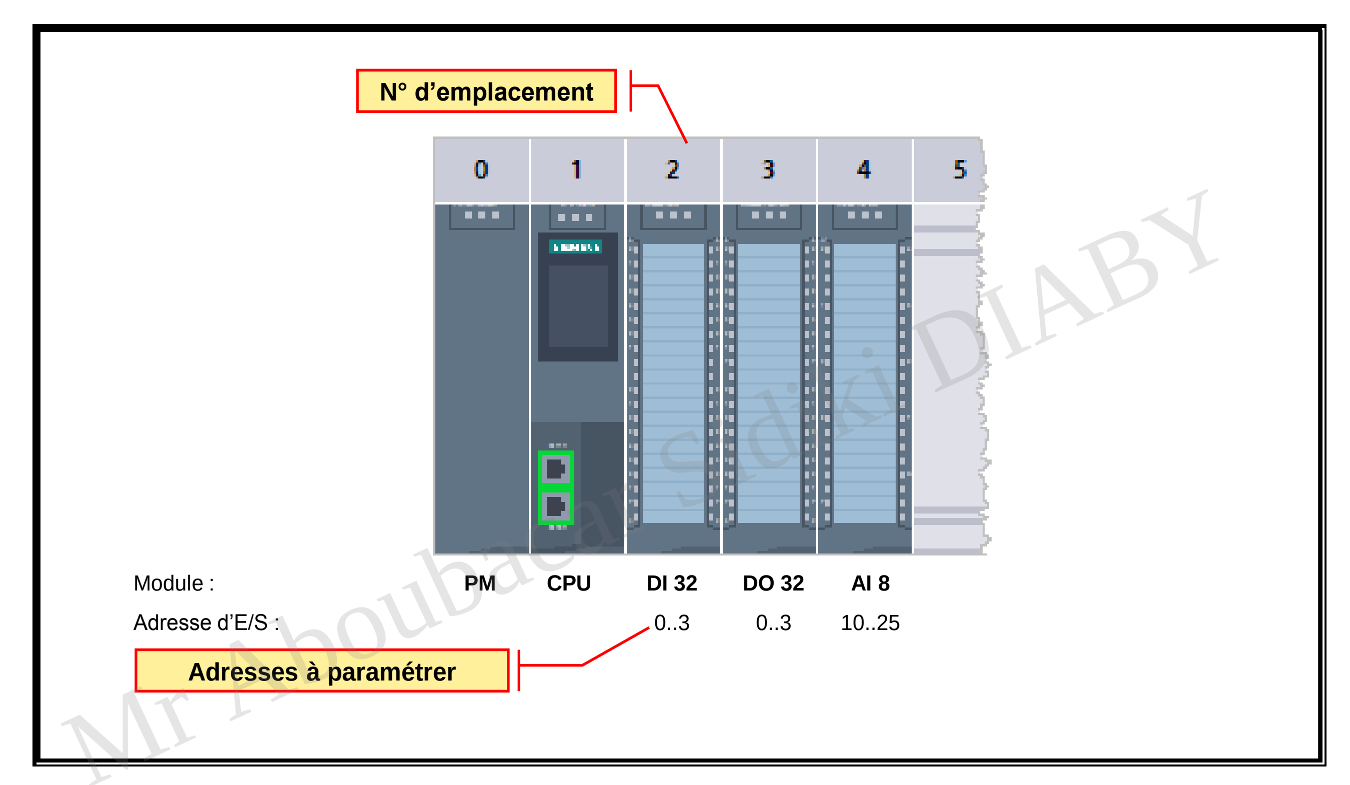


Composition du poste de travail

Le poste de travail est constitué des éléments suivants :

- Automate S7-1500 avec E/S centralisées,
- I/O-Device (ET200SP) avec E/S décentralisées
- Pupitre Opérateur permettant le contrôle commande de l'installation (IHM),
- Un tableau de commande pour signaux TOR avec boutons et LED
- Des E/S Analogiques avec 2 curseurs et 2 afficheurs.

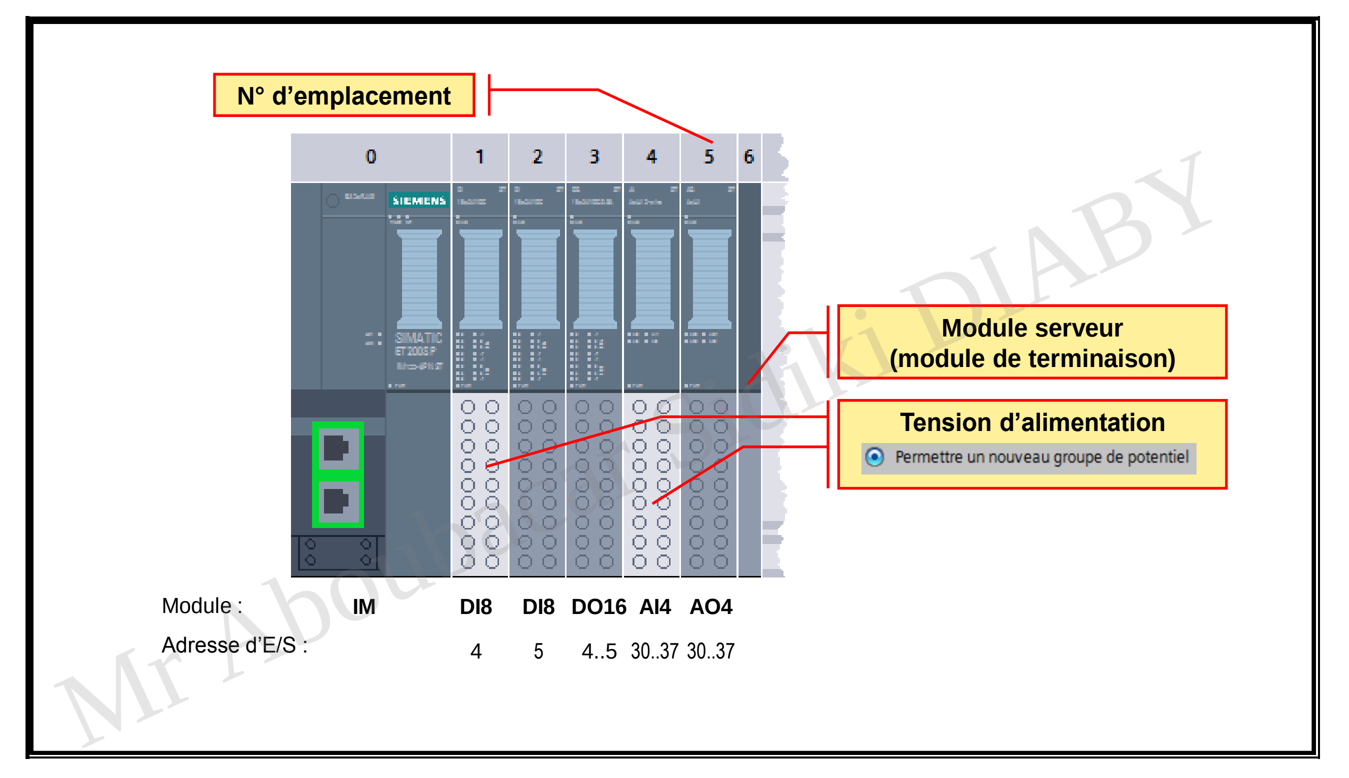
1.2.1. Configuration matérielle de l'automate S7-1500



On dispose de deux modules TOR de 32 voies comme périphérie centralisée. L'adressage doit s'effectuer à partir de l'adresse = 0.

Comme des voies TOR sont par ailleurs encore disponibles sur la périphérie décentralisée, l'adressage du module analogique de la périphérie centralisée doit commencer à l'adresse = 10.

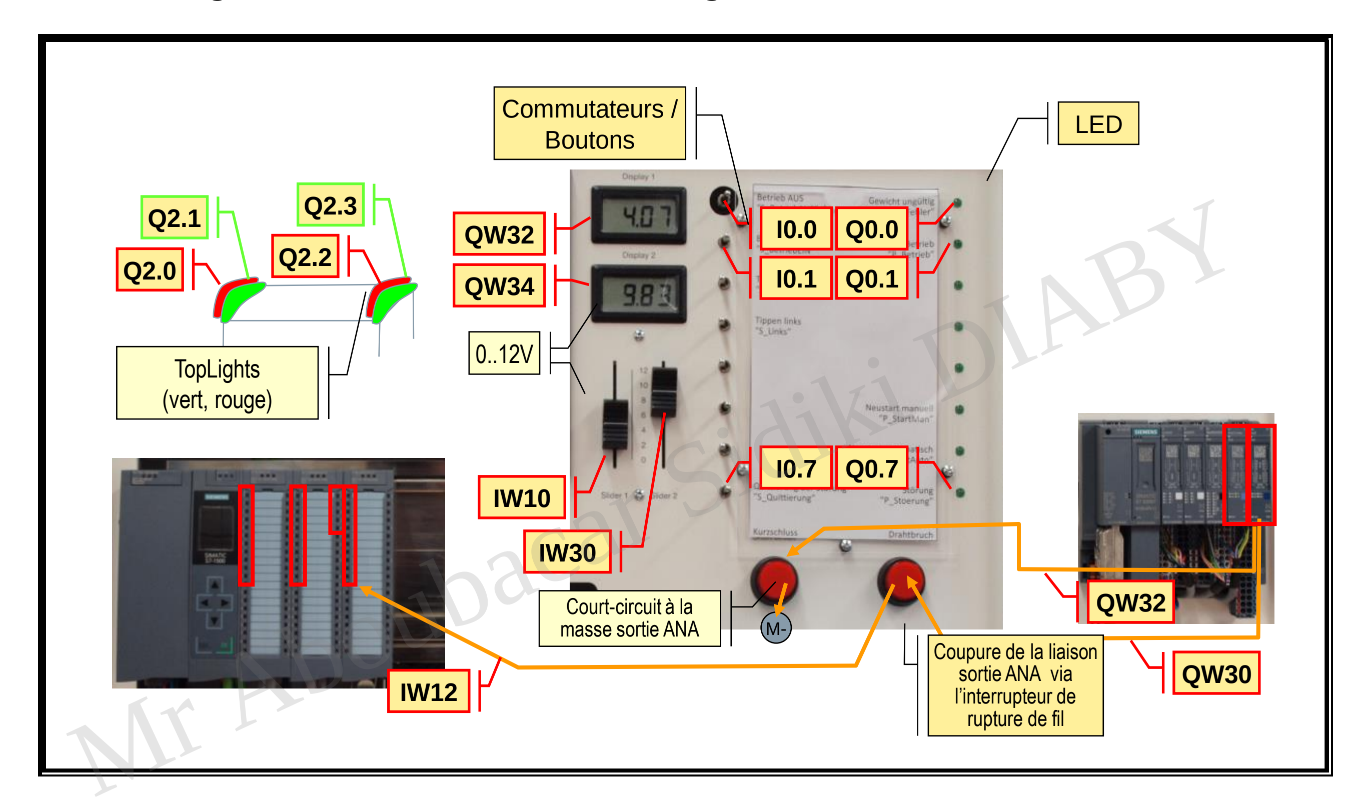
1.2.2. Configuration matérielle de la périphérie décentralisée ET200SP



On dispose de trois modules TOR 16 voies comme périphérie décentralisée. Dans la zone d'adressage, ceux-ci doivent se raccorder à la périphérie TOR centralisée à partir de l'adresse = 4.

La périphérie analogique décentralisée doit commencer à l'adresse = 30.

1.2.3. Organes de commande et d’affichage du matériel de formation



Organes de commande

En plus du pupitre opérateur à écran tactile, on dispose d’organes de commande séparés pour la commande de l’installation :

- 8 commutateurs ou boutons
- 2 potentiomètres pour l’entrée de consignes ou la simulation de signaux d’entrée analogiques
- Un interrupteur de rupture de fil coupant la liaison SANA1 périphérie décentralisée avec EANA2 périphérie centralisée
- Un interrupteur de court-circuit mettant en court-circuit à la masse la SANA2 de la périphérie décentralisée

Organes d’affichage

En plus du pupitre opérateur à écran tactile, on dispose d’organes d’affichage séparés pour la visualisation des informations du processus :

- 8 LED
- 2 afficheurs de tension numériques pour l’affichage des signaux de sortie analogiques
- Au-dessus de l’appareil de formation se trouvent, à droite et à gauche, 2 rangées de LED, dites « TopLights » (2 vertes, 2 rouges). Celles-ci peuvent être commandées via 4 STOR.

Raccordement / Câblage

Ces organes de commande et d'affichage sont aussi bien raccordés à la périphérie centralisée qu'à la périphérie décentralisée.

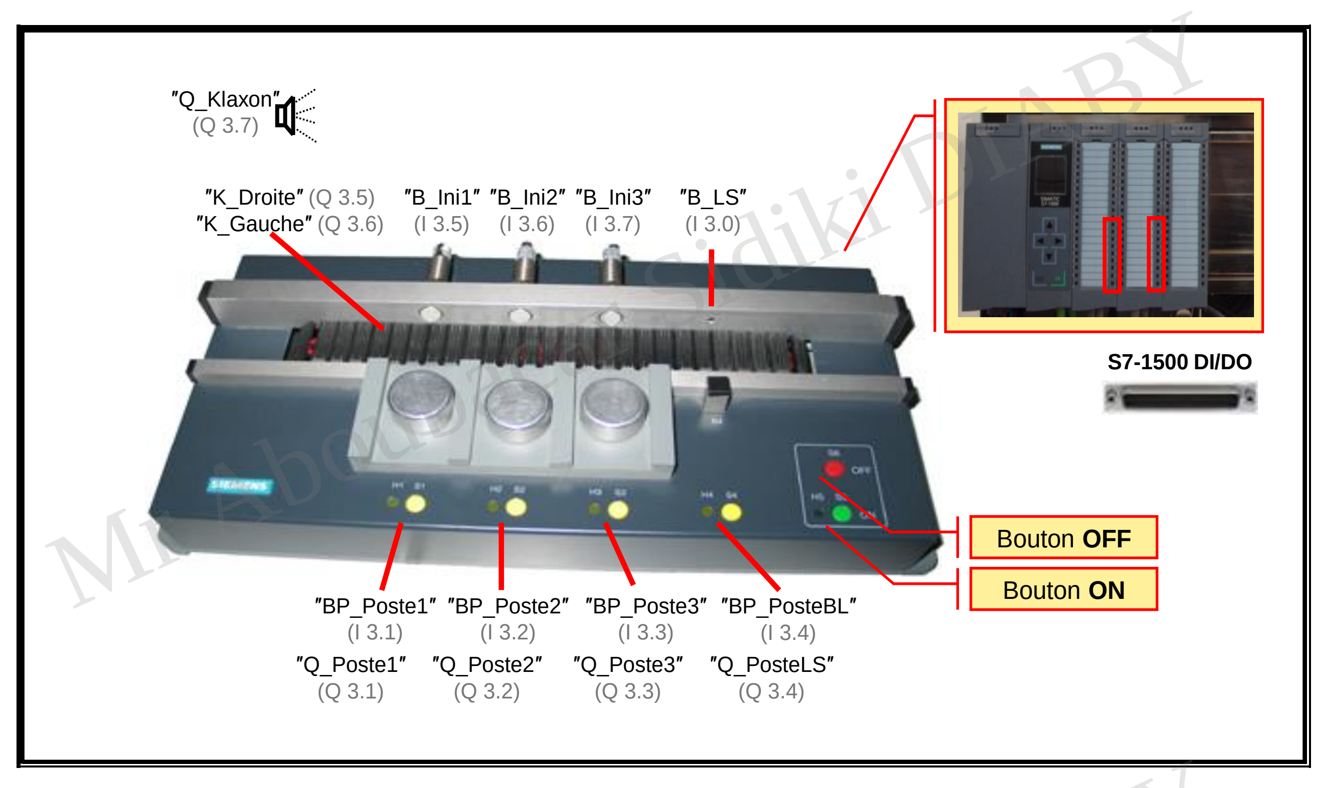
Voie	ETOR centralisée	STOR centralisée	EANA central.	
0	Commutateur/bouton 1	LED 1	Curseur 1 (gauche)	
1	Commutateur/bouton 2	LED 2	Entrée rupture de fil de SANA de la périphérie décentralisée ET200SP	
2	Commutateur/bouton 3	LED 3	-	
3	Commutateur/bouton 4	LED 4	-	
4	Commutateur/bouton 5	LED 5	-	
5	Commutateur/bouton 6	LED 6	-	
6	Commutateur/bouton 7	LED 7	-	
7	Commutateur/bouton 8	LED 8	-	
...				
16	-	TopLight rouge gauche		
17	-	TopLight verte gauche		
18	-	TopLight rouge droite		
19	-	TopLight verte droite		
...				
	ETOR décentralisée	**STOR décentralisée**	**EANA décentr.**	**SANA décentr.**
0	-	-	Curseur 2 (droite)	Sortie rupture de fil sur EANA de la périphérie décentralisée S7-1500
1	-	-	-	Afficheur tension 1
2	-	-	-	Afficheur tension 2
...				

Adressage

Pour obtenir l'adressage indiqué dans la figure, il faut définir les paramètres d'adressage correspondants des modules dans l'outil « Configuration des appareils ».

1.3. Configuration et raccordement de la maquette de bande transporteuse

1.3.1. Raccordement à la périphérie TOR du S7-1500



Deux points de raccordement pour la maquette de bande transporteuse sont prévus en face arrière de la valise de formation.
→ Utiliser le raccordement S7-1500 DI/DO

Raccordement / Câblage

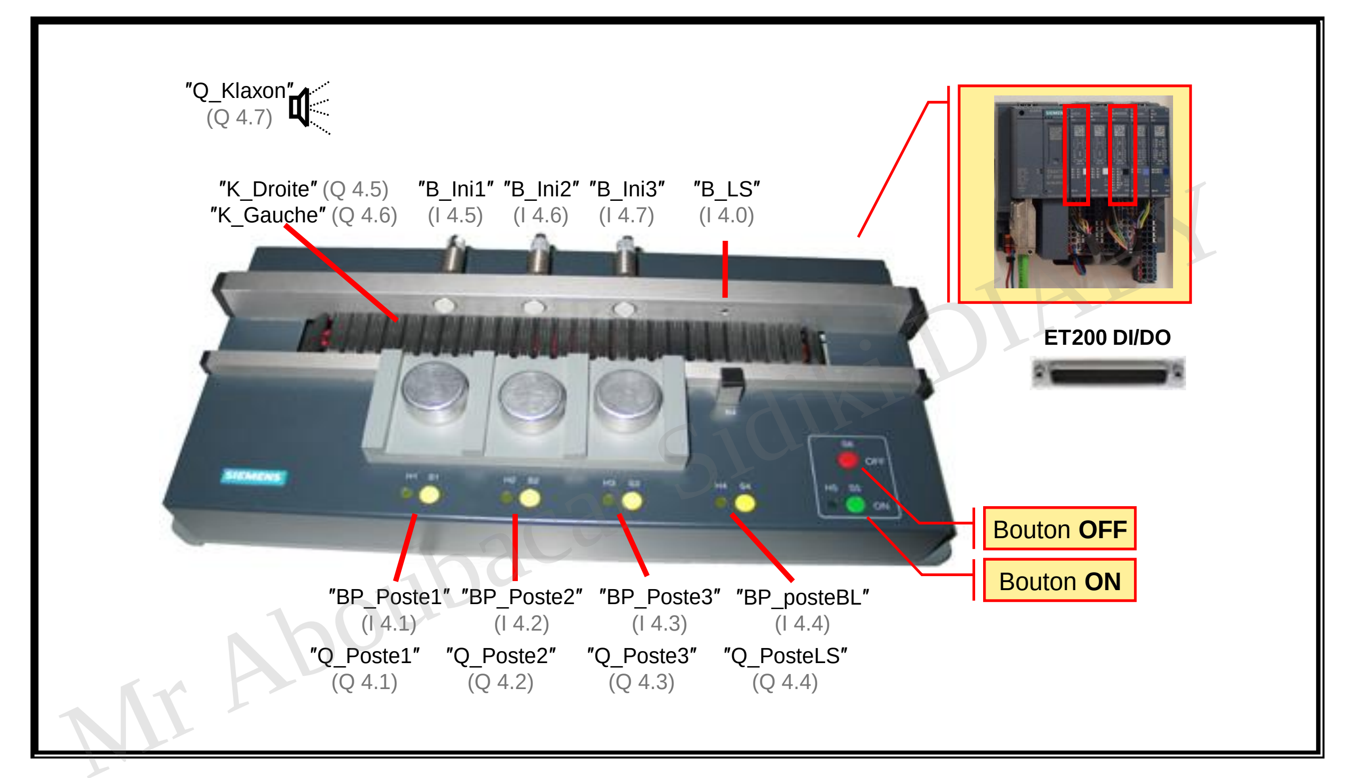
La maquette de bande transporteuse est raccordée à la périphérie centralisée.

Voie	ETOR centralisée		STOR centralisée	
24	Barrière lumineuse	I3.0	-	
25	Bouton Poste 1	I3.1	Lampe Poste 1	Q3.1
26	Bouton Poste 2	I3.2	Lampe Poste 2	Q3.2
27	Bouton Poste 3	I3.3	Lampe Poste 3	Q3.3
28	Bouton Poste BL	I3.4	Lampe Poste BL	Q3.4
29	Détecteur INI1	I3.5	Marche droite	Q3.5
30	Détecteur INI2	I3.6	Marche gauche	Q3.6
31	Détecteur INI3	I3.7	Klaxon	Q3.7

Adressage

Pour obtenir l'adressage indiqué dans la figure ci-dessus, il faut définir les paramètres d'adressage correspondants des modules avec l'outil « Configuration des appareils ».

1.3.2. Raccordement à la périphérie TOR de l'ET200SP



Deux points de raccordement pour la maquette de bande transporteuse sont prévus en face arrière de la valise de formation.
→ Utiliser le raccordement ET200 DI/DO

Raccordement / Câblage

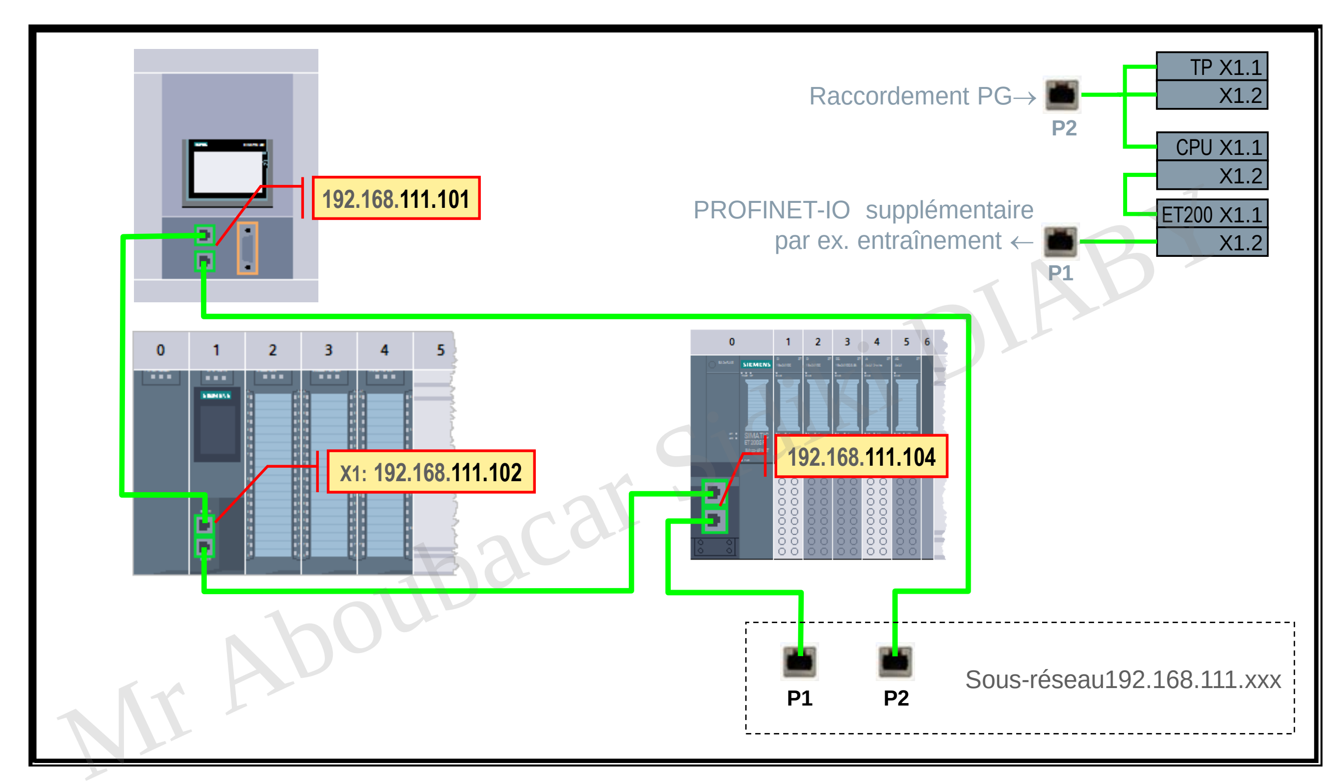
La maquette de bande transporteuse est raccordée à la périphérie décentralisée de l'ET200SP.

Voie	ETOR de l'ET200S		STOR de l'ET200S	
0	Barrière lumineuse	I4.0	-	
1	Bouton Poste 1	I4.1	Lampe Poste 1	Q4.1
2	Bouton Poste 2	I4.2	Lampe Poste 2	Q4.2
3	Bouton Poste 3	I4.3	Lampe Poste 3	Q4.3
4	Bouton Poste BL	I4.4	Lampe Poste BL	Q4.4
5	Détecteur INI1	I4.5	Marche à droite	Q4.5
6	Détecteur INI2	I4.6	Marche à gauche	Q4.6
7	Détecteur INI3	I4.7	Klaxon	Q4.7

Adressage

Pour obtenir l'adressage indiqué dans la figure ci-dessus, il faut définir les paramètres d'adressage correspondants des modules avec l'outil « Configuration des appareils ».

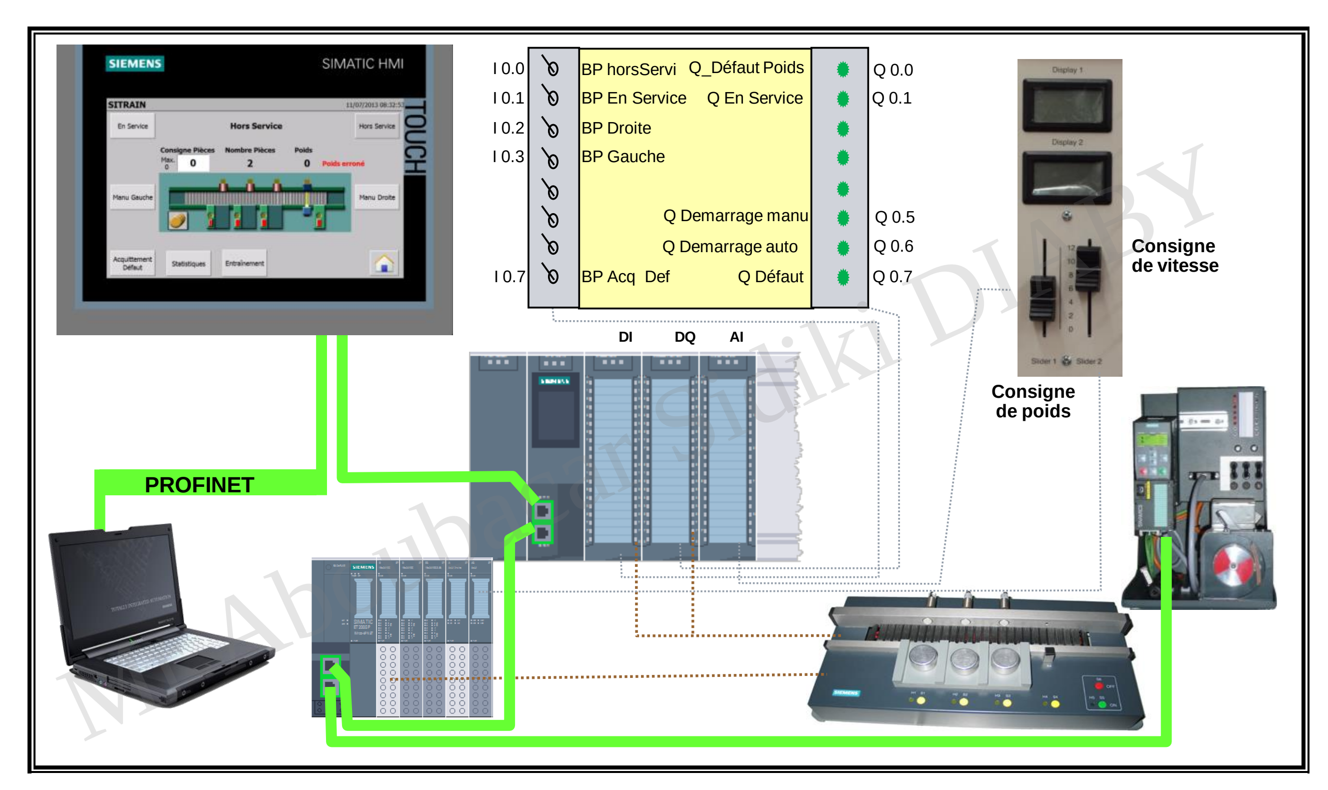
1.4. Câblage et adresses IP des modules



Deux sous-réseaux Ethernet sont câblés sur la valise de formation.

- Sous-réseau 192.168.111.xxx
  Prise de raccordement « P3 » ↔ TP-X1 ↔ CPU-X1 ↔ ET200SP-X1 ↔ Prise de raccordement « P2 »
- Sous-réseau 192.168.112.xxx
  Prise de raccordement « P1 » ↔ CPU-X2

1.5. Poste de travail avec convoyeur à bande et pupitre à écran tactile



Le convoyeur à bande sert à transporter des pièces. Il peut être commandé selon deux modes. Pour sélectionner le mode, utilisez d'abord les boutons du simulateur, puis les boutons correspondants du pupitre opérateur à écran tactile.

Hors service « Q En Service » (Q0.1) = désactivé

Le moteur du convoyeur peut à présent être commandé par impulsions vers la droite ou vers la gauche. Pour ce faire, utilisez d'abord les boutons du simulateur, puis les boutons correspondants du pupitre opérateur à écran tactile.

Commandez ensuite l'entraînement comme s'il devait entraîner la bande. La consigne de vitesse du moteur doit être entrée à l'aide d'un paramètre du variateur.

En service « Q En Service » (Q0.1) = activé

Les pièces sont transportées par le convoyeur du poste 1 ou 2 vers la droite jusqu'à la barrière lumineuse.

Si la durée du transport dépasse 6 secondes, le moteur du convoyeur est automatiquement arrêté et le défaut est signalé sur le simulateur ainsi que sur le pupitre opérateur à écran tactile. Une nouvelle opération de transport ne peut être lancée qu'après acquittement du défaut à l'aide du bouton du simulateur ou de l'écran tactile.

En cours de service (« Q En Service » = activé), toutes les pièces qui parviennent jusqu'à la barrière lumineuse sont comptées et pesées.
La consigne de poids est entrée à l'aide du curseur de gauche « Curseur1 ». Les poids des pièces sont mémorisés dans une mémoire de données. Lorsque le nombre de consigne (entré sur le pupitre à écran tactile) de poids de pièces est atteint, la LED du poste barrière lumineuse (« Q_PosteLS ») de la maquette commence à clignoter à une fréquence de 1 Hz.
Une nouvelle opération de transport ne peut être lancée qu'après acquittement de la signalisation à l'aide du bouton (« BP_posteBL ») du poste barrière lumineuse.

Les LED de visualisation des postes 1 et 2...

- restent allumées en continu lorsqu’une nouvelle pièce peut être placée sur la bande transporteuse ;
- clignotent à une fréquence de 1 Hz au niveau du poste où la présence d’une pièce a été détectée par le détecteur de proximité correspondant, mais uniquement tant que la bande n’a pas encore démarré ;
- clignotent à une fréquence de 2 Hz tant que le moteur de la bande est en marche.

La LED du poste barrière lumineuse

- clignote à une fréquence de 2 Hz tant que le moteur de la bande est en marche.
- reste allumée en continu lorsque le nombre de consigne défini a été atteint.

L’entraînement sera ensuite commandé en supplément comme s’il entraînait la bande. La vitesse du moteur peut être réglée à l’aide du curseur de droite « Curseur2 ».

Sommaire 2

2. Vue d'ensemble du système

A l'issue du chapitre, vous allez ...



... connaître le concept « Totally Integrated Automation » (T.I.A.)

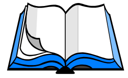
... Connaître le TIA Portal information center

... avoir un aperçu de la nouvelle famille SIMATIC S7-1200/1500

... avoir un aperçu des différents modules

... Connaître le système d'automatisation S7-1200/1500 avec l'afficheur et la SIMATIC Mémory Card

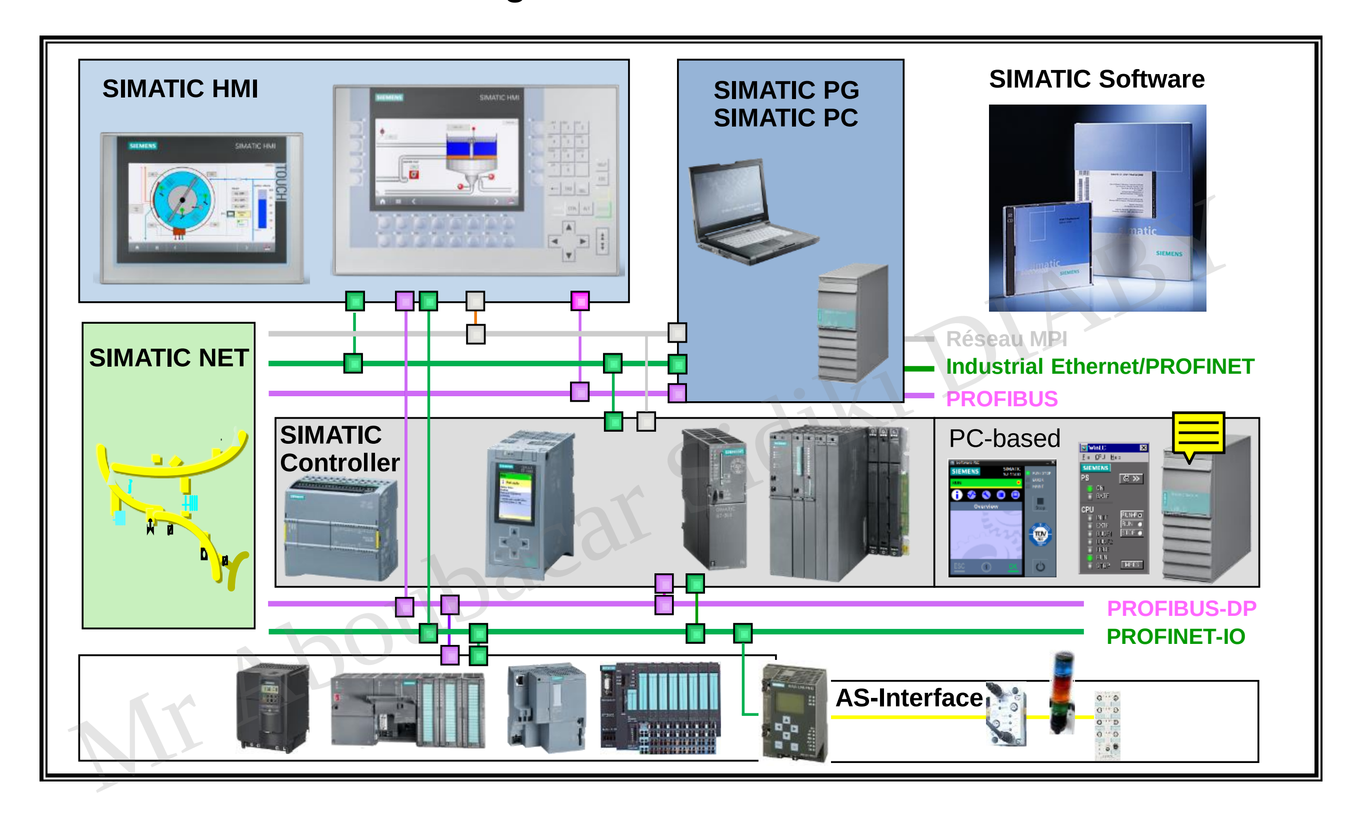
… Connaitre les systèmes d'automatisation S7-300 et S7-400



Ce chapitre présente la gamme des équipements d'automatisme S7.

A l'issue de ce chapitre, le participant au stage de formation connaîtra le concept T.I.A. Il aura une vue d'ensemble des modules utilisés sur un API S7.

2.1. Présentation de la gamme SIMATIC



INTRODUCTION

Outre une alimentation en énergie, l'exploitation des machines, d'équipements et de processus industriels sur un site de production requiert généralement des systèmes de commande capables d'assurer le fonctionnement, le contrôle, la surveillance et l'arrêt des installations.

De la logique câblée à l'automate programmable

Avant l'émergence des automates programmables, la logique des programmes de commande était déterminée par le câblage des contacteurs et des relais dans les armoires électriques. Le schéma de câblage était adapté en fonction de la tâche à exécuter. On parlait alors de logique câblée. Aujourd'hui, les tâches d'automatisation sont assurées par des automates programmables. La logique stockée dans la mémoire programme de l'automate est indépendante de la configuration matérielle et du câblage. Elle peut donc être modifiée à tout moment à l'aide d'une console de programmation.

Automatisation entièrement intégrée

Aujourd'hui, les opérations de production ne sont plus traitées isolément. Elles sont intégrées à un processus global de production. L'intégration de l'environnement d'automatisation est assurée désormais grâce à :

- une configuration et une programmation homogènes des différentes unités du système
- une gestion cohérente des données
- une communication globale entre les équipements d'automatisme.

2.2. Centre d’information TIA Portal



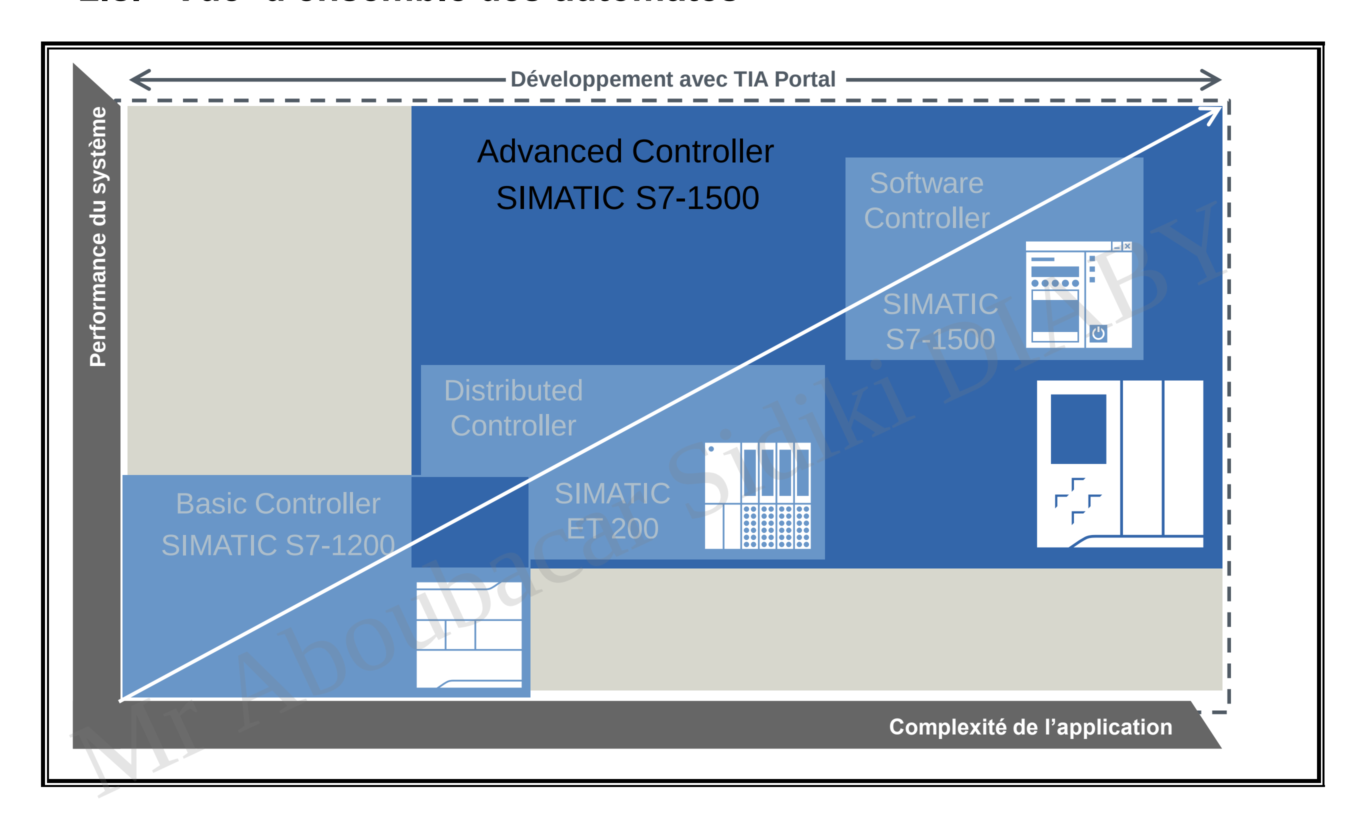
Avec le numéro de la contribution 65601780, vous accédez à la page de démarrage « TIA-Portal – Aperçu des documents et des liens les plus importants ».

Vous trouverez un point d’entrée pour tous les documents et les liens importants concernant TIA Portal ainsi que les automates S7-1200 et S7-1500.

De plus vous accédez au « Centre d’information TIA Portal » par lequel vous obtenez également les liens et les informations importantes.

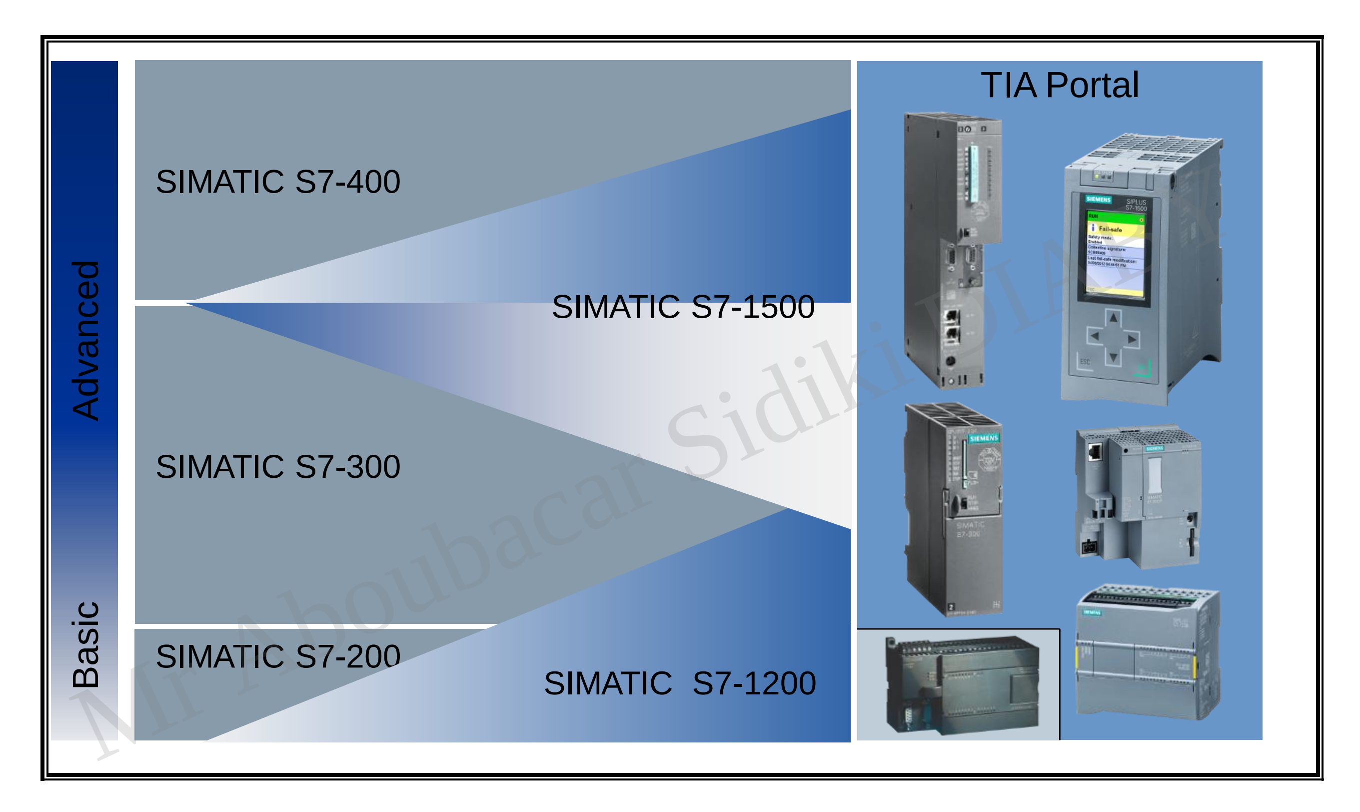
Mr Aboubacar Sidiki DIABY

2.3. Vue d'ensemble des automates



En fonction de la complexité de la tâche, vous pouvez utiliser différents automates allant du S7-1200 jusqu'au S7-1500.

2.4. Positionnement des automates modulaires S7

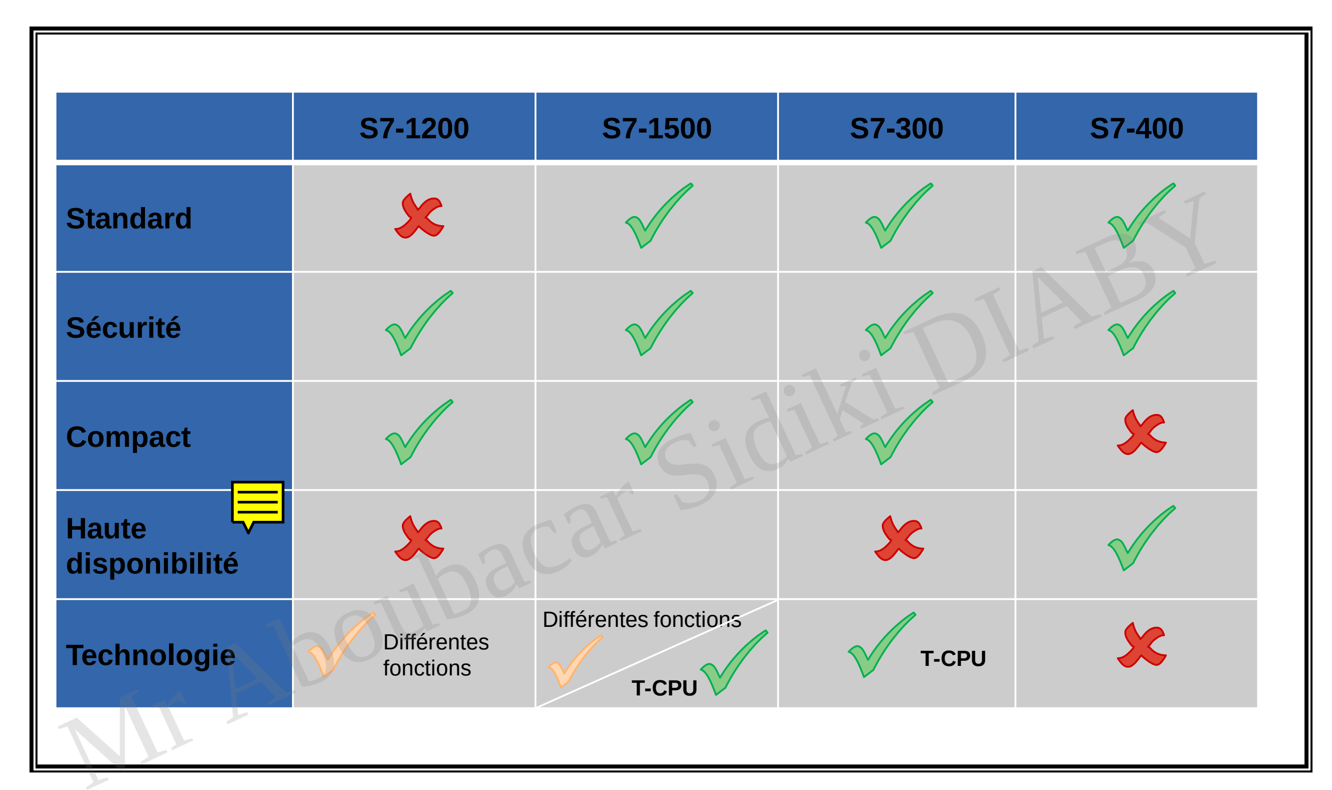


SIMATIC S7

Les automates programmables de la gamme se répartissent dans la classe de puissance micro-API (S7-1200) et la classe de puissance moyenne à supérieure (S7-1500).

La classe de puissance inférieure à moyenne utilisée jusqu'ici (S7-300) et la classe de puissance moyenne à supérieure (S7-400) seront couvertes à l'avenir par le système S7-1500.

2.4.1. Vue d'ensemble des modules centralisés disponibles

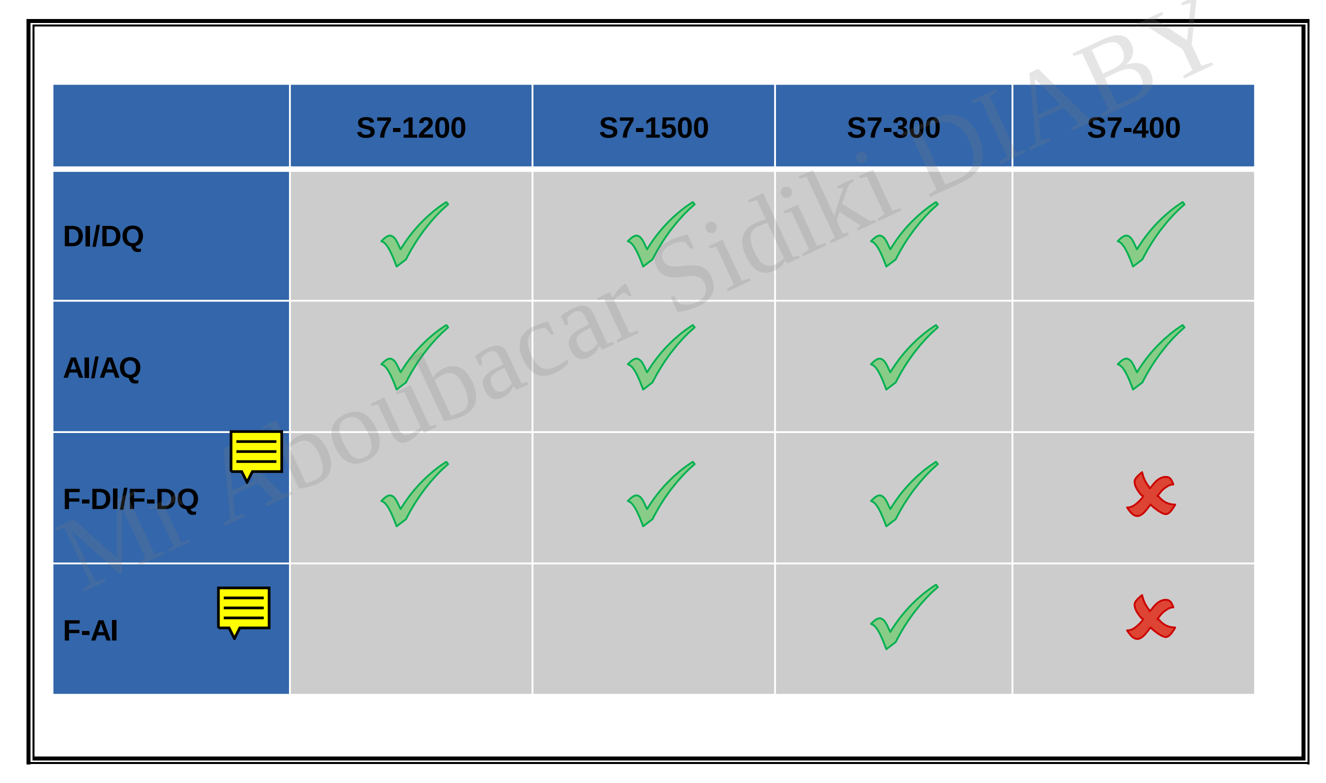


	S7-1200	S7-1500	S7-300	S7-400
Standard	✗	✓	✓	✓
Sécurité	✓	✓	✓	✓
Compact	✓	✓	✓	✗
Haute disponibilité	✗		✗	✓
Technologie	✓ Différentes fonctions	Différentes fonctions ✓ / T-CPU ✓	✓ T-CPU	✗

Plus d'informations disponibles en suivant le lien :

TIA Portal Information Center → Informations produits → Contrôleurs

2.4.2. Vue d'ensemble des modules de signaux disponibles (centralisés)



	S7-1200	S7-1500	S7-300	S7-400
DI/DQ	✓	✓	✓	✓
AI/AQ	✓	✓	✓	✓
F-DI/F-DQ	✓	✓	✓	✗
F-AI			✓	✗

Plus d'informations disponibles en suivant le lien :

TIA Portal Information Center → Informations produits → Contrôleurs

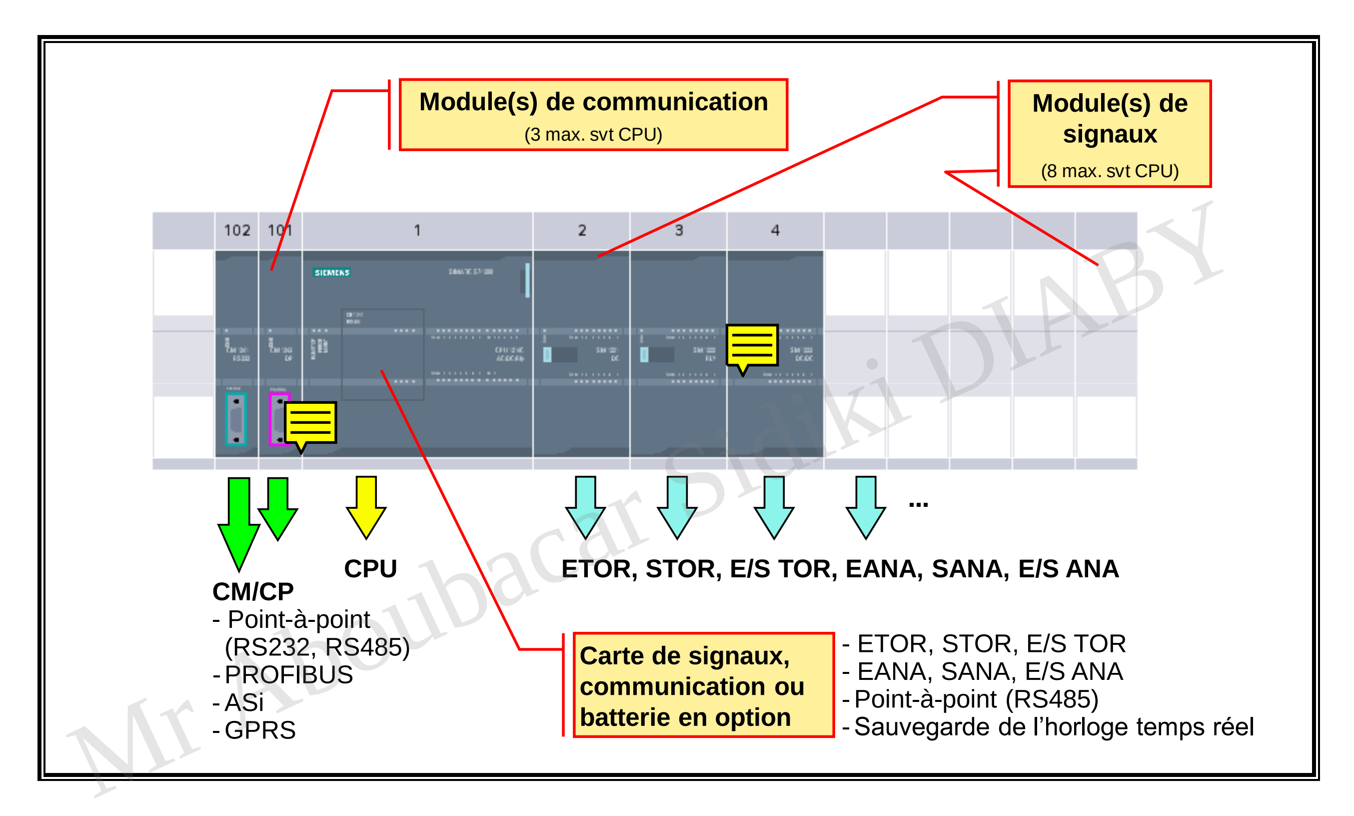
2.5. SIMATIC S7-1200 : Micro-automate modulaire



Caractéristiques

- Micro-automate modulaire pour applications exigeants une faible puissance de calcul
- Gamme échelonnée de CPU
- Vaste gamme de modules
- Capacité d'extension jusqu'à 8 modules (en fonction de la CPU)
- Mise en réseau possible avec PROFIBUS ou PROFINET
- Règles de montage des modules
  - Modules de communication (CM) à gauche de la CPU (nombre en fonction de la CPU)
  - Modules de signaux (SM) à droite de la CPU (nombre en fonction de la CPU)
- Automatisme complet avec CPU et E/S dans un seul appareil
  - E/S tout-ou-rien et analogiques
  - une extension avec carte de signaux
- Micro-automate avec fonctions intégrées

2.5.1. SIMATIC S7-1200 : Modules



Règles de montage des modules

- Modules de communication (CM) à gauche de la CPU (nombre en fonction de la CPU)
- Modules de signaux (SM) tout-ou-rien et analogiques à droite de la CPU (nombre en fonction de la CPU)

Modules de signaux

- ETOR, STOR, E/S TOR (24V=, relais)
- EANA, SANA ou E/S ANA (tension, courant, résistance, thermocouple)

Modules de communication (CM - Communication module, CP – Communication processor)

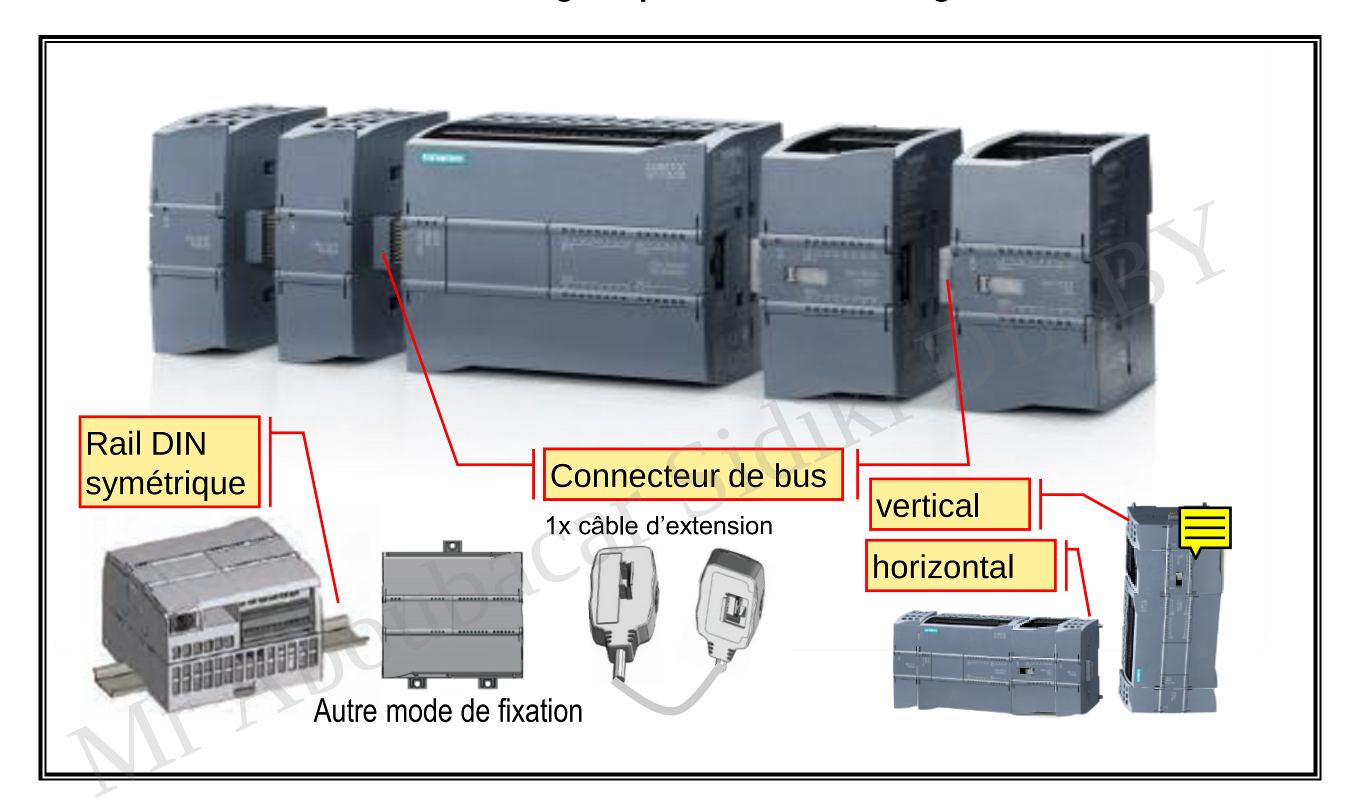
- Couplage point-à-point (RS232, RS485)
- PROFIBUS
- Maître ASi
- Contrôle à distance (fonctionnalité GPRS)

Carte d'extension

Permet d'étendre la CPU avec une périphérie embarquée ou une interface

La carte batterie assure la sauvegarde à long terme de l'horloge temps réel.

2.5.2. SIMATIC S7-1200 : Montage et positions de montage



Montage

Les modules peuvent être montés sur un rail DIN symétrique ou fixés par vis dans une armoire.

Câble d'extension S7-1200 (longueur 2 m)

Assure une flexibilité accrue lors de la configuration du système S7-1200. Un seul câble d'extension est autorisé par système CPU

- soit entre la CPU et le premier SM, soit entre deux SM

Connecteur de bus

Situé sur le côté gauche des modules SM sous forme de connecteur insérable

Situé mécaniquement sur le côté droit des CM/CP

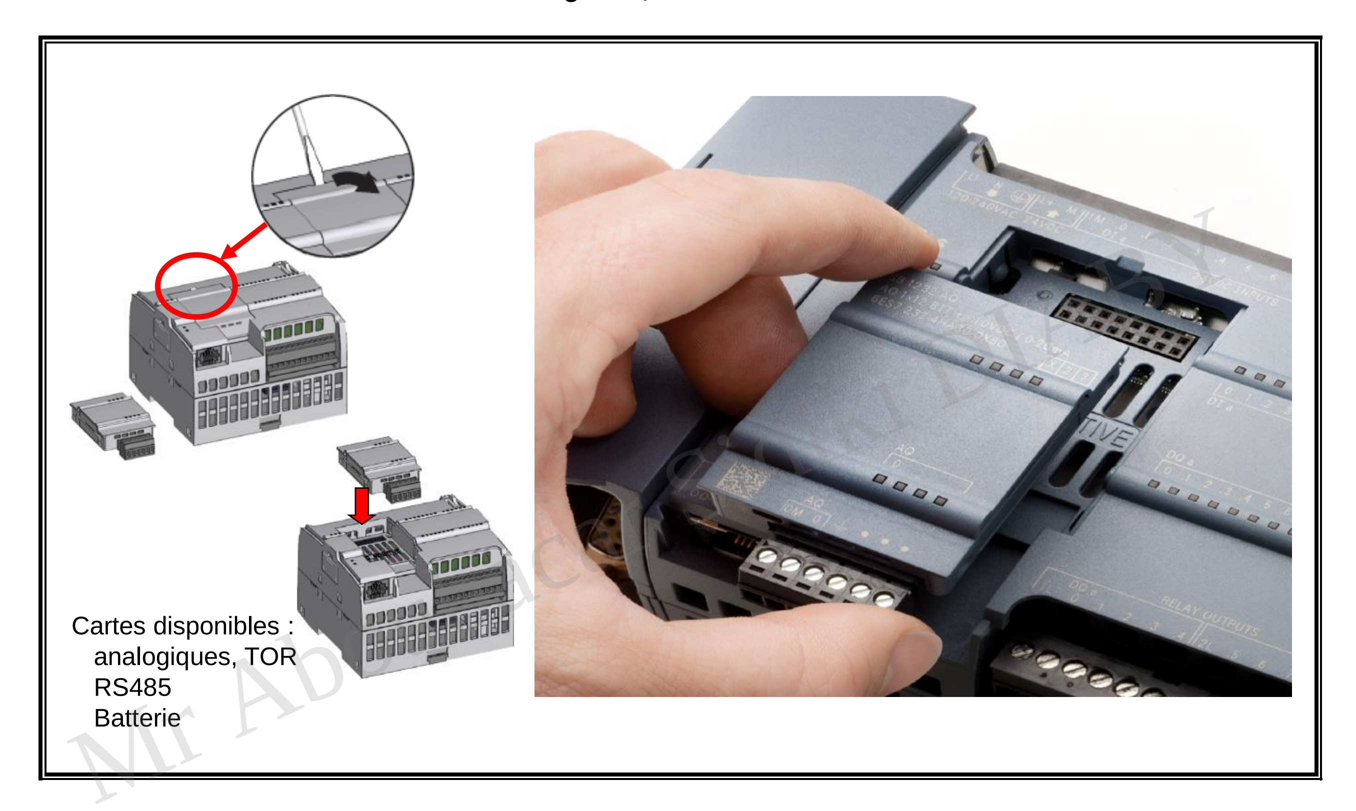
Positions de montage

Les appareils peuvent être montés horizontalement comme verticalement.



Attention !
En cas de montage vertical, la température ambiante max. admissible est diminuée de 10 °C.

2.5.2.1. SIMATIC S7-1200 : Cartes de signaux, de communication et batterie



Utilisation

Ces cartes permettent d'adapter la CPU aux exigences des applications développées pour l'installation. La taille de la CPU reste inchangée après le montage de ces cartes.

Carte de signaux (SB)

- Carte de signaux TOR
  - uniquement entrées
  - uniquement sorties
  - entrées et sorties
- Carte de signaux analogique
  - uniquement entrées
  - uniquement sorties

Carte de communication (CB)

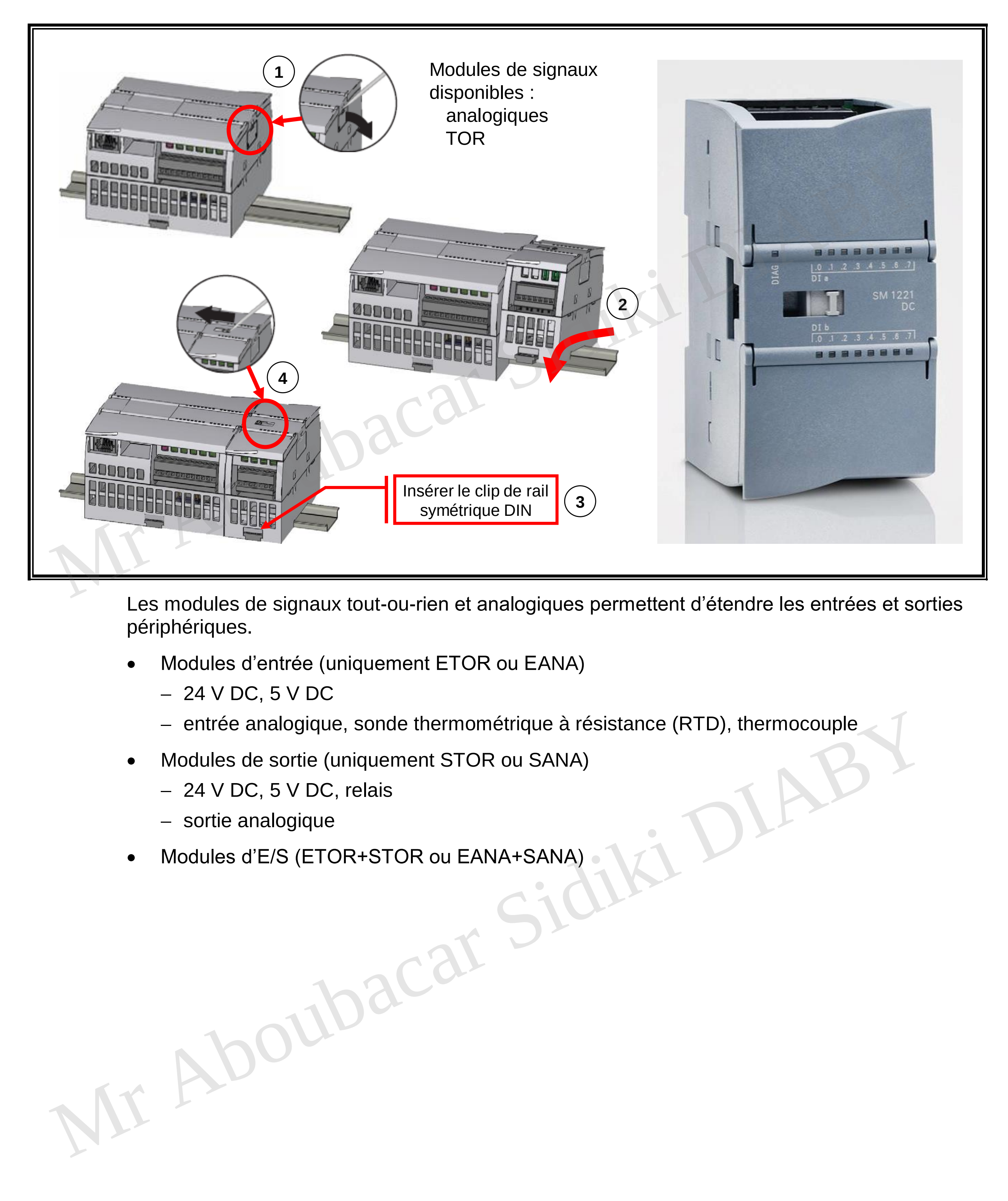
- Interface RS485

Carte batterie (BB)

La carte batterie (Battery Board) fonctionne avec une pile CR1025. Elle assure la sauvegarde à long terme de l'horloge temps réel.

- Durée de sauvegarde sans carte batterie : 20 jours en moyenne / min. 12 jours à 40 °C
- Durée de sauvegarde avec carte batterie : env. 1 an

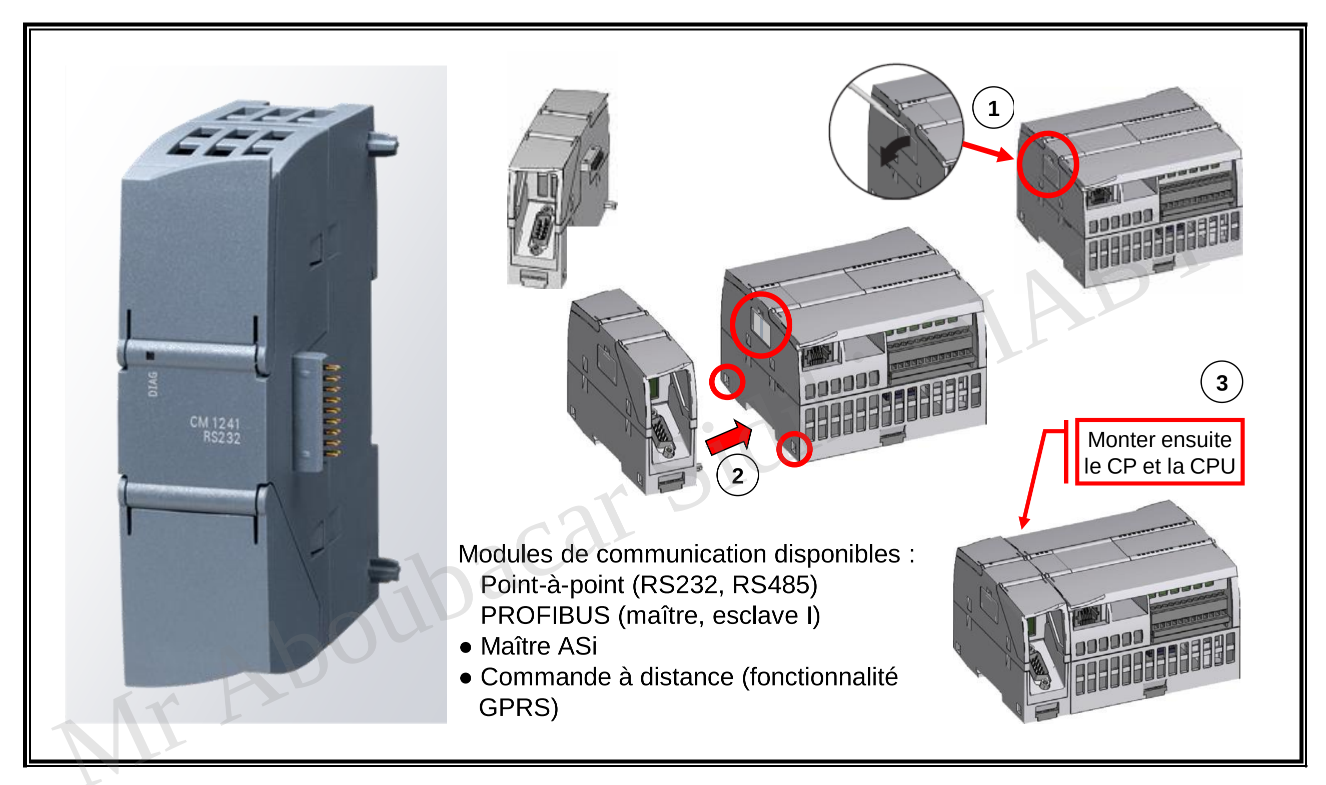
2.5.2.2. SIMATIC S7-1200 : Modules de signaux



Les modules de signaux tout-ou-rien et analogiques permettent d'étendre les entrées et sorties périphériques.

- Modules d'entrée (uniquement ETOR ou EANA)
  - 24 V DC, 5 V DC
  - entrée analogique, sonde thermométrique à résistance (RTD), thermocouple
- Modules de sortie (uniquement STOR ou SANA)
  - 24 V DC, 5 V DC, relais
  - sortie analogique
- Modules d'E/S (ETOR+STOR ou EANA+SANA)

2.5.2.3. SIMATIC S7-1200 : Modules de communication



Pour étendre les interfaces, on dispose de modules de communication (CM - Communication module, CP - Communication processor).

- Couplage point-à-point (RS232)
- Couplage point-à-point (RS485)
- Maître PROFIBUS DP
- PROFIBUS esclave I
- Maître ASi
- Contrôle à distance (fonctionnalité GPRS)

2.5.3. SIMATIC S7-1200 : Accessoires



Bloc d'alimentation

Pour alimenter les différentes variantes de CPU en courant continu 24 V

- Entrée : 120 / 230 V AC 50 / 60 Hz, 1,2 / 0,7 A
- Sortie : 24 V DC / 2,5 A

Commutateur Ethernet compact (Compact Switch Module)

Les commutateurs Ethernet servent à la réalisation de réseaux de communication.

- 4 x prises RJ 45 10 / 100 Mbit/s

Carte mémoire

Une carte mémoire SIMATIC est une carte mémoire SD préformatée par Siemens, qui ne doit en aucun cas être reformatée avec une application Windows. Pour pouvoir mémoriser et effacer des données sur la carte SD, il faut désactiver la protection en écriture en faisant coulisser la languette de protection sur le côté de la carte.

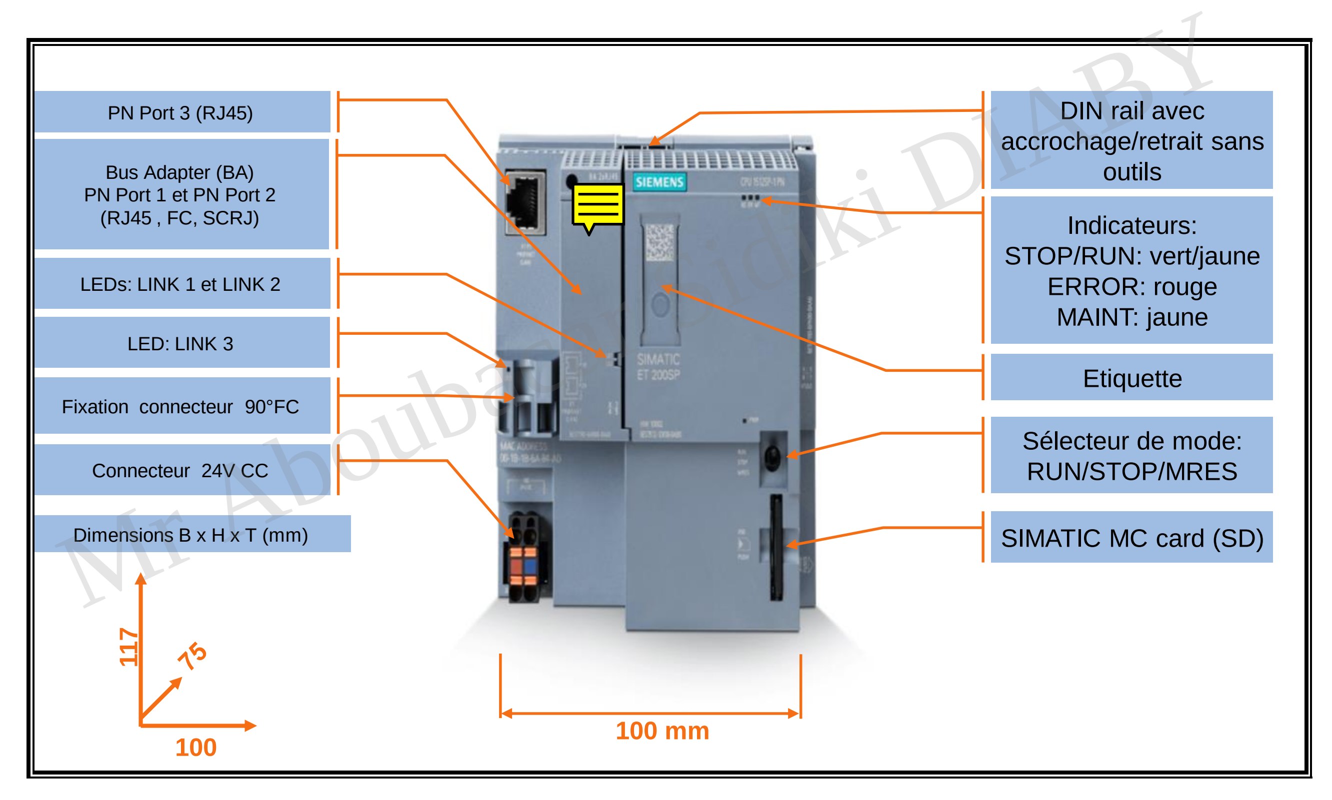
- La carte mémoire s'utilise...
  - ... pour transférer un programme sur une ou plusieurs CPU
  - ... pour mettre à jour le firmware
  - ... en remplacement de la mémoire de chargement interne de la CPU
- La carte mémoire permet de mémoriser...
  - ... des programmes S7
  - ... des données utilisateur quelconques (par ex. la documentation)
  - ... des projets

2.5.4. SIMATIC S7-1500 / ET 200 SP CPUs

Contrôleurs ET200SP et ET200pro



ET200SP – Vue d'ensemble

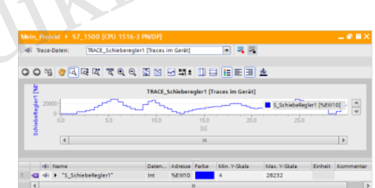


2.6. SIMATIC S7-1500 : Automate modulaire pour applications de complexité moyenne à élevée



Points forts du système SIMATIC S7-1500

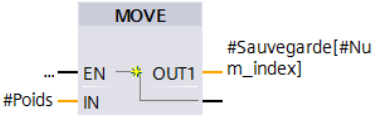
- performance maximale du système (borne à borne)
  - Temps d'exécution du programme CPU extrêmement court
  - Bus interne performant
  - Interface PROFINET et PROFINET IO IRT sur chaque CPU
  - Diagnostic système automatiquement activé jusqu'au niveau du canal d'E/S
- Trace pour toutes les variables de la CPU



- Afficheur sur la CPU pour:
  - Accéder aux MLFB, version firmware et numéro de série
  - Mise en service : (par exemple : affecter l'adresse IP, le nom de la station)
  - Sauvegarder / Restaurer
  - Diagnostiquer



- Programmation simplifiée par des instructions conviviales pour les langages CONT/LOG/LIST



Caractéristiques

- Système modulaire pour applications exigeant une puissance de calcul moyenne à élevée
- Gamme échelonnée de CPU
- Afficheur pour paramétrages de base de la CPU (heure système, interfaces...) et pour la visualisation des informations de diagnostic et d'état (tampon de diagnostic, messages, état de la CPU...)
- Vaste gamme de modules
- Bus périphérique performant pour un raccordement efficace du processus via la périphérie centrale
- Capacité d'extension jusqu'à 32 modules par châssis

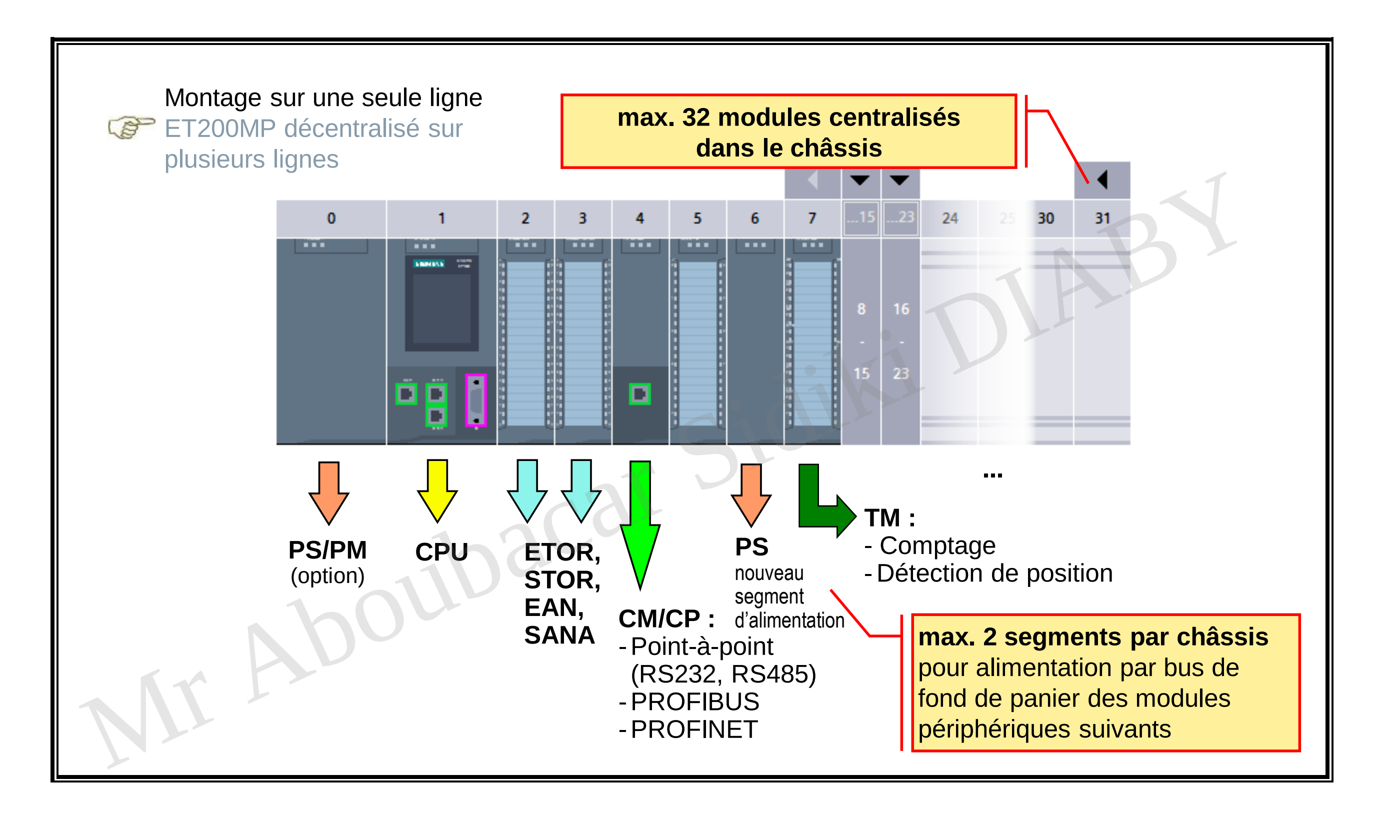
  ☞ Un seul châssis possible actuellement. Configuration multichâssis via ET200MP décentralisé
- Mise en réseau possible via PROFIBUS ou PROFINET
- Règles de montage des modules
  - à gauche de la CPU :
    1x alimentation (PM ou PS)
  - à droite de la CPU :
    Modules de signaux (tout-ou-rien, analogiques), modules technologiques, modules de communication et alimentations supplémentaires
- Pas de règles de montage spécifique pour les modules à droite de la CPU

2.6.1. SIMATIC S7-1500 : Portfolio



Mr Aboubacar Sidiki DIABY

2.6.2. SIMATIC S7-1500 : Modules



Règles de montage des modules

- 1x PS/PM à l'emplacement 0
- 1x CPU à l'emplacement 1
- Disposition quelconque à partir de l'emplacement 2

Modules de signaux

- Modules d'entrée TOR : 24V=, 230V~
- Modules de sortie TOR : 24V=, 230V~
- Modules d'entrée analogiques : tension, courant, résistance, thermocouple
- Modules de sortie analogiques : tension, courant

Modules de communication (CP - Communication processor, CM - Communication module)

- Couplage point-à-point
- PROFIBUS
- PROFINET

☞ Les CP et CM sont tous deux des modules de communication.
Les CP disposent généralement de plus de fonctionnalités que les CM (par ex. serveur Web propre, pare-feu, etc.).

Modules technologiques (TM - Technology module)

- Comptage
- Détection de position

Alimentation

Les modules périphériques montés dans le châssis centralisé du S7-1500 nécessitent une alimentation système via le bus de fond de panier (liaison de communication avec la CPU) et une alimentation externe (alimentation des circuits d'entrée et sortie des capteurs et actionneurs).

- **PM - Power module → Alimentation externe**
  alimente en 24 V DC les circuits d'entrée et sortie des modules et les capteurs et actionneurs

   Si la CPU est alimentée en 24 V via une alimentation externe (PM), elle fournit une alimentation système 12 W aux premiers modules périphériques enfichés.

- **PS - Power system → Alimentation système**
  alimente les modules S7-1500 du châssis centralisé via le bus de fond de panier

   Chaque CPU fournit une alimentation système 12 W aux premiers modules périphériques enfichés. Des segments d'alimentation supplémentaires doivent éventuellement être configurés en fonction des modules périphériques utilisés.

   Une alimentation système (PS) peut également alimenter le circuit de charge des modules 24 V DC en plus de la CPU.

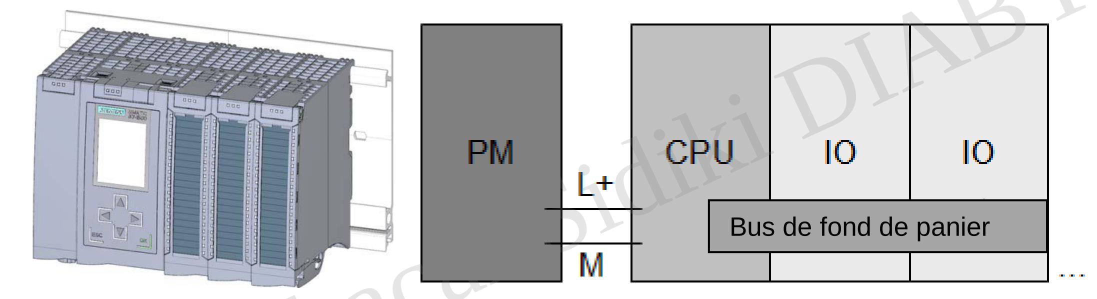
Alimentation et segments d'alimentation des modules périphériques

Avec des configurations plus importantes ou des configurations avec des modules périphériques à plus forte consommation dans le châssis centralisé (généralement en cas d'utilisation de CP, CM, TM), il convient de créer des segments d'alimentation.

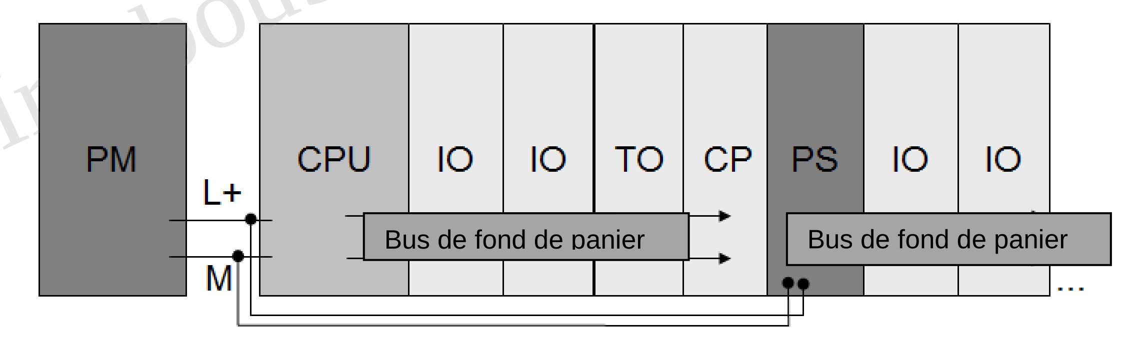
Il est possible de créer au maximum 3 segments d'alimentation par châssis (1 x segment CPU plus 2 supplémentaires).

Dans la variante de configuration avec segments d'alimentation supplémentaires, on utilise des modules d'alimentation système supplémentaires (PS) à droite de la CPU. La CPU continue de commander tous les modules du châssis. On ne segmente ici que l'alimentation système des modules périphériques.

Exemple de petite configuration S7-1500

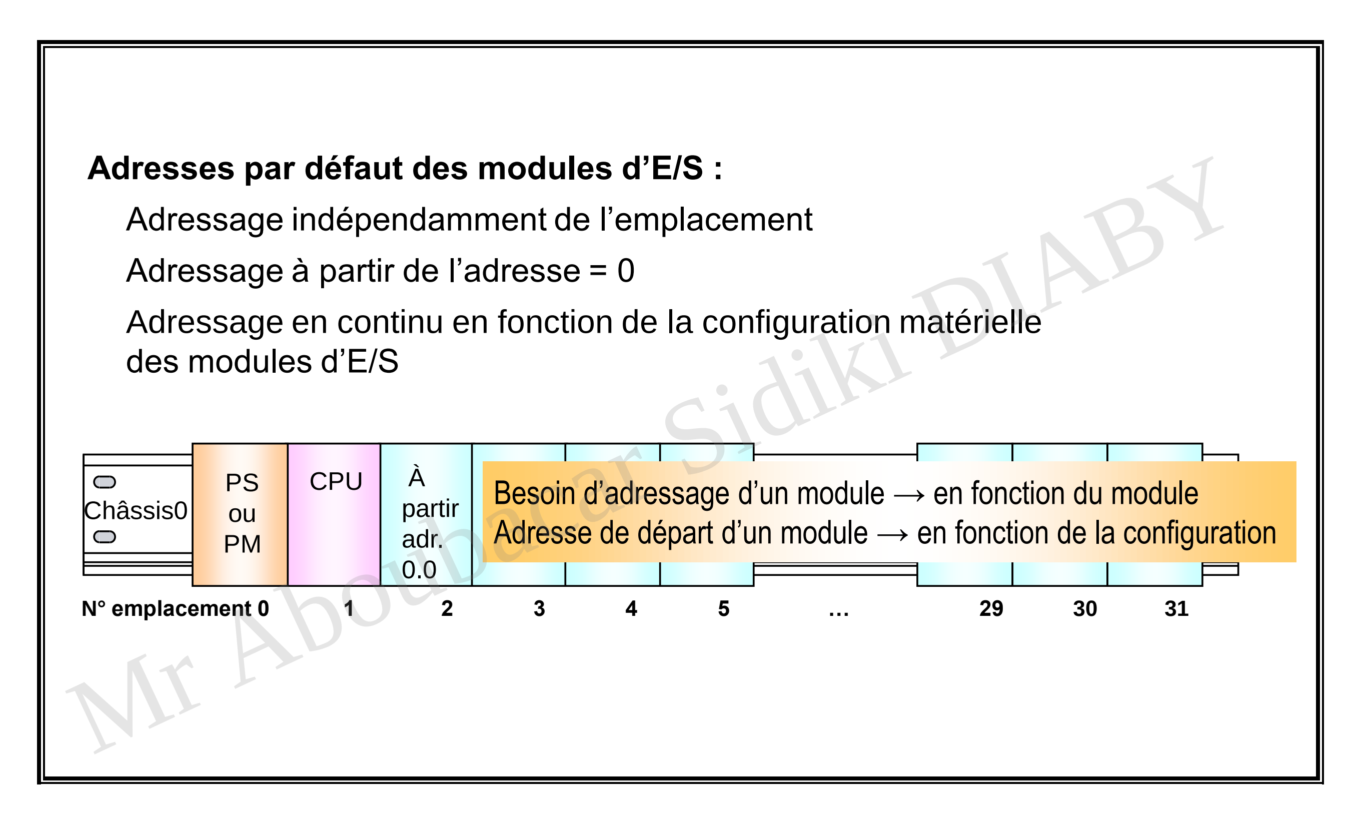


Exemple de configuration S7-1500 avec un 2e segment d'alimentation



Couplage de châssis d'extension
Il n'est pas prévu de configuration multichâssis centralisée. Une extension est possible via le système périphérique décentralisé ET200MP.

2.7. Adressage des E/S du S7-1500 pour une configuration à un seul châssis



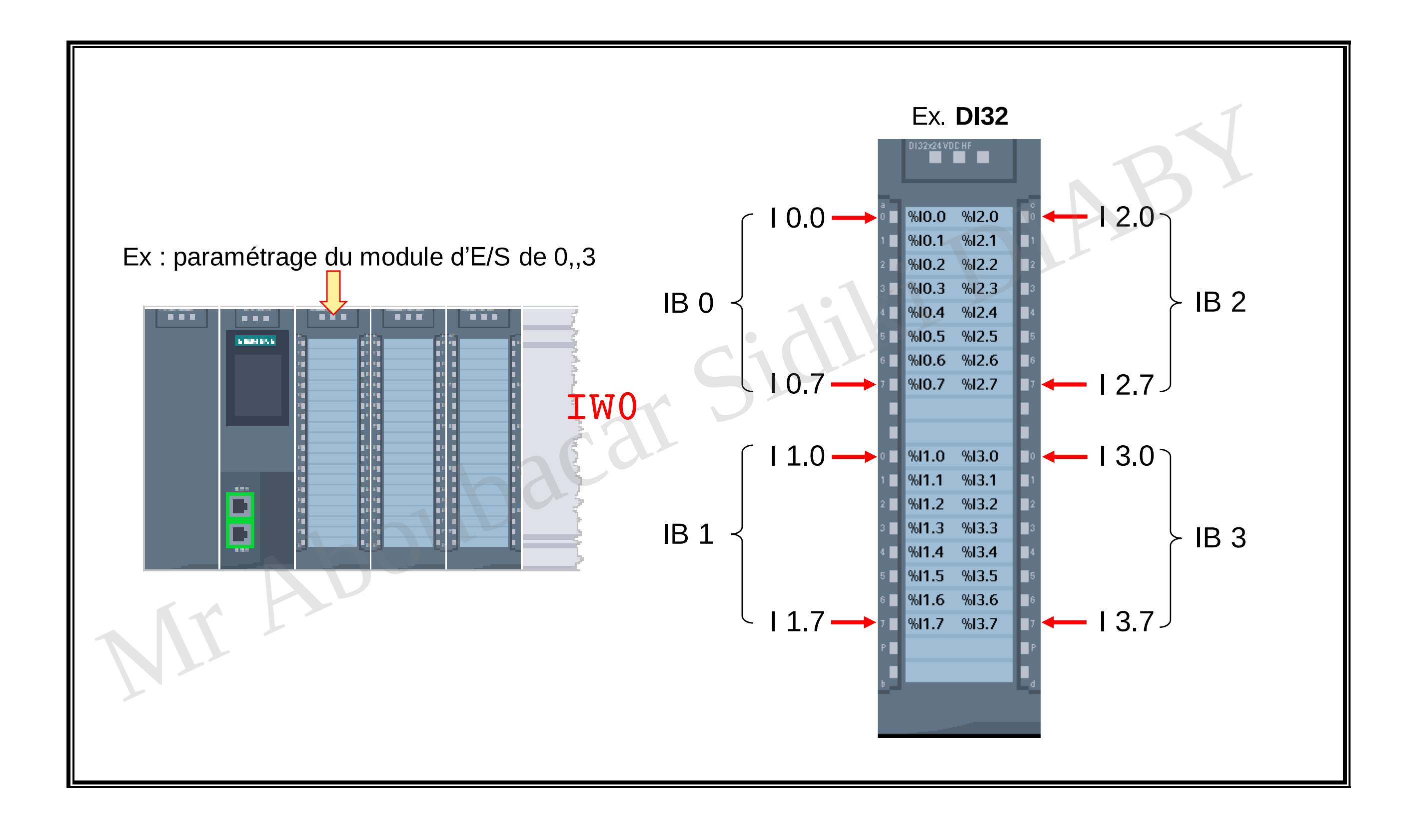
Pour pouvoir être utilisé par le programme utilisateur, chaque module périphérique se voit affecter une « adresse de module » permettant d'adresser les différentes voies (canaux) et la zone de données d'état/de commande. En accédant en lecture ou en écriture à ces zones d'adressage, le programme utilisateur peut interagir avec la périphérie du processus qui est raccordée aux modules.

Adressage par défaut des emplacements

L'affectation d'adresses en fonction de l'emplacement n'existe que pour le S7-300 et le S7-400.

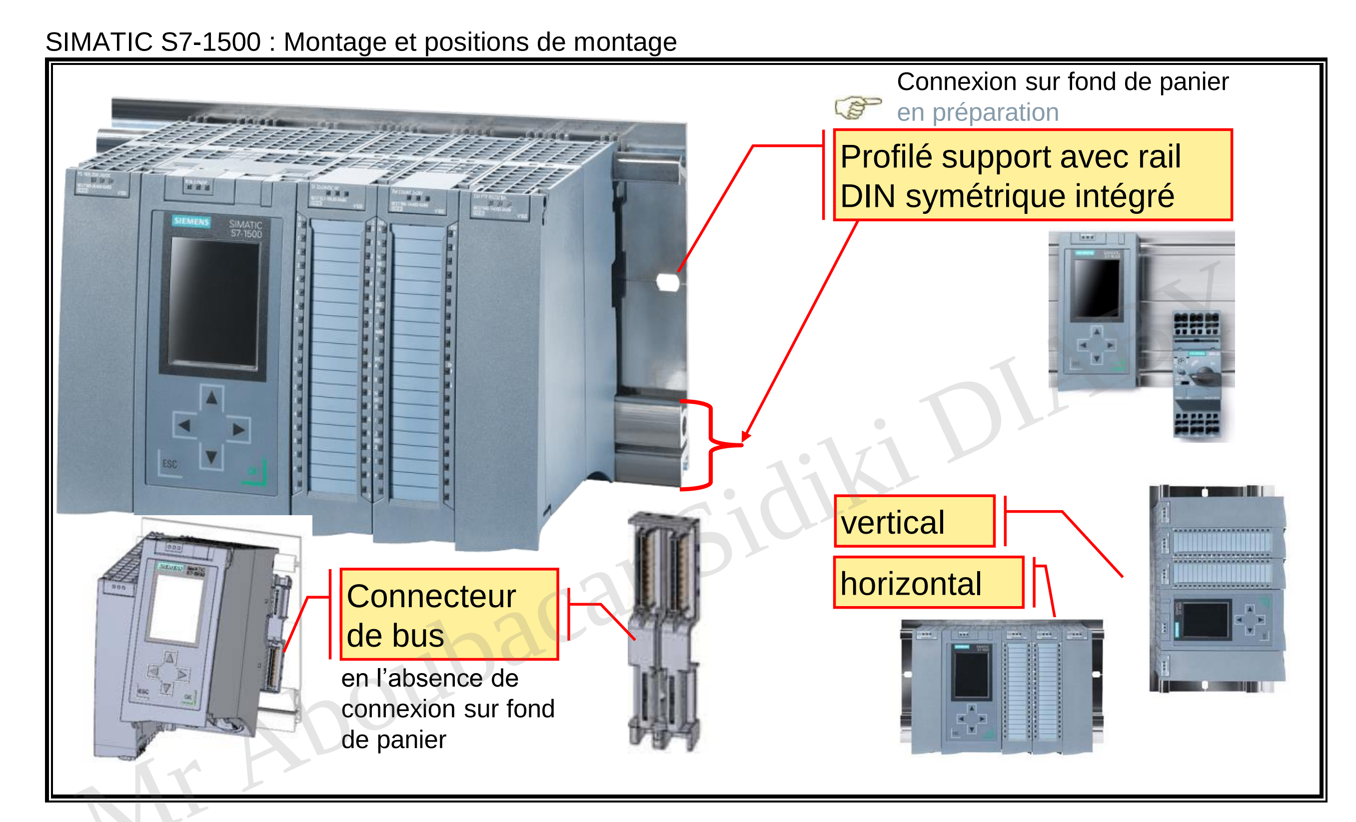
Lors de la configuration du S7-1500, le système propose, pour chaque module ajouté, une affectation d'adresse sous forme **d'affectation continue de la zone d'adressage de la périphérie** à partir de l'adresse = 0.

2.7.1.1. Adressage des modules S7-1500 TOR





SIMATIC S7-1500 : Montage et positions de montage



Montage Le rail esT différent de celui du 300

Les modules sont montés sur un profilé support S7-1500.

Connecteur de bus

En cas de montage sur profilé support, les modules sont reliés à l'aide d'un connecteur en U.

Le connecteur en U réalise la connexion mécanique et électrique entre les modules. Il est compris dans la fourniture de tous les modules périphériques.

Positions de montage

Les appareils peuvent être montés horizontalement comme verticalement.

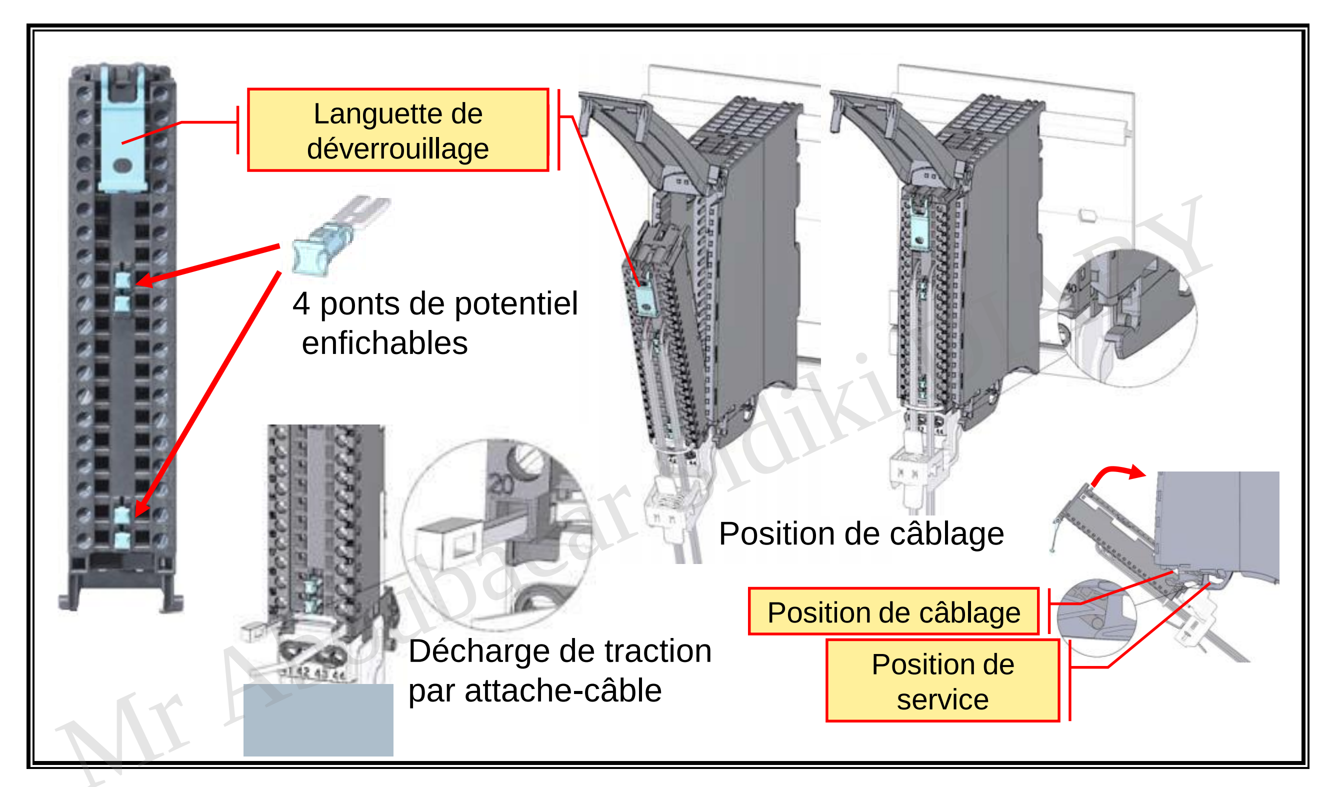


Attention !
En cas de montage vertical, la température ambiante max. admissible est diminuée de 20 °C (température ambiante 0..40 °C).

Rail DIN symétrique intégré

La CPU S7-1500 dispose d'un profilé support doté d'un rail DIN symétrique dans sa partie inférieure. Ce rail permet de monter directement à côté du S7-1500 des composants conformes à EN 60715 comme des bornes, des disjoncteurs de ligne, des petits contacteurs ou des composants similaires.

2.7.1.2. SIMATIC S7-1500 : Technique de raccordement/Connecteur frontal



Soulevez la languette de déverrouillage pour libérer le connecteur frontal de sa position de blocage et l'extraire du module.

Caractéristiques du connecteur frontal

- 40 points de raccordement
- Techniques de raccordement:
  - borne à vis
  - SIMATIC TOP connect
    Câblage système pour le raccordement de capteurs et actionneurs
    → connecteur frontal S7-1500 câblé avec 20 ou 40 fils individuels (préfabriqué)

Position de précâblage

Dans cette position, le connecteur frontal est encliqueté en haut du capot frontal. Il dépasse encore du module. Le connecteur frontal et le module ne sont pas encore reliés électriquement.

Ponts de potentiel

Les ponts de potentiels simplifient la réalisation de groupes de charge de même potentiel. Ils peuvent être enfichés à quatre positions.
Il n'y a qu'une seule liaison de gauche à droite !

Codage automatique des modules périphériques

Cette technique permet un remplacement rapide et sûr du connecteur frontal.

2.7.2. SIMATIC S7-1500 : Afficheur de la CPU → Vue d'ensemble



Toutes les CPU S7-1500 sont livrées avec un afficheur

Deux tailles en fonction du type de CPU
1,36“ jusqu'à CPU1513
2,4“ à partir de CPU1516

Possèdent leur propre MLFB
→ commande possible comme pièce de rechange

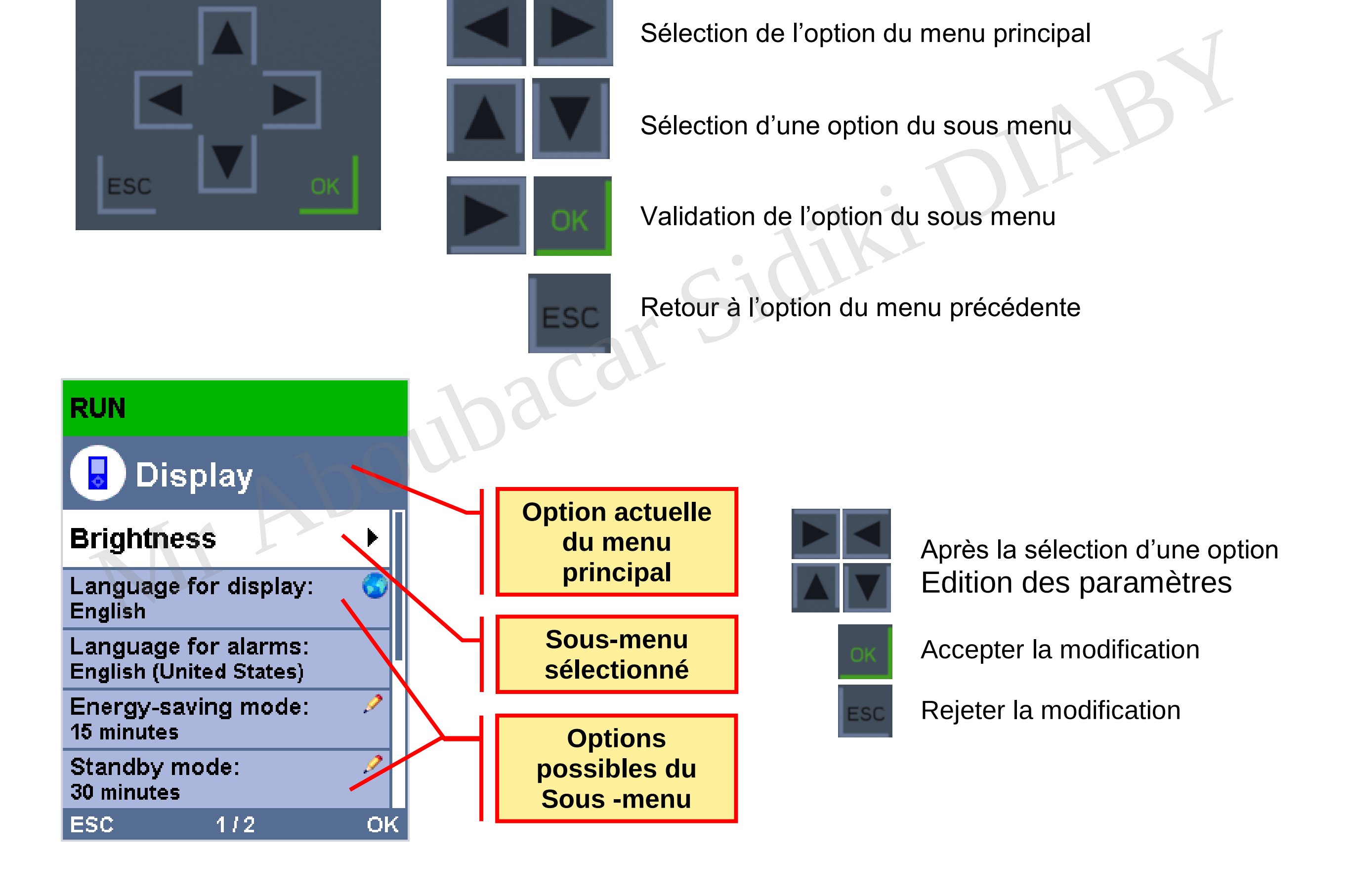
La CPU est exploitable sans afficheur (autre volet frontal)

Retrait et enfichage possibles en cours de fonctionnement → la CPU reste en RUN

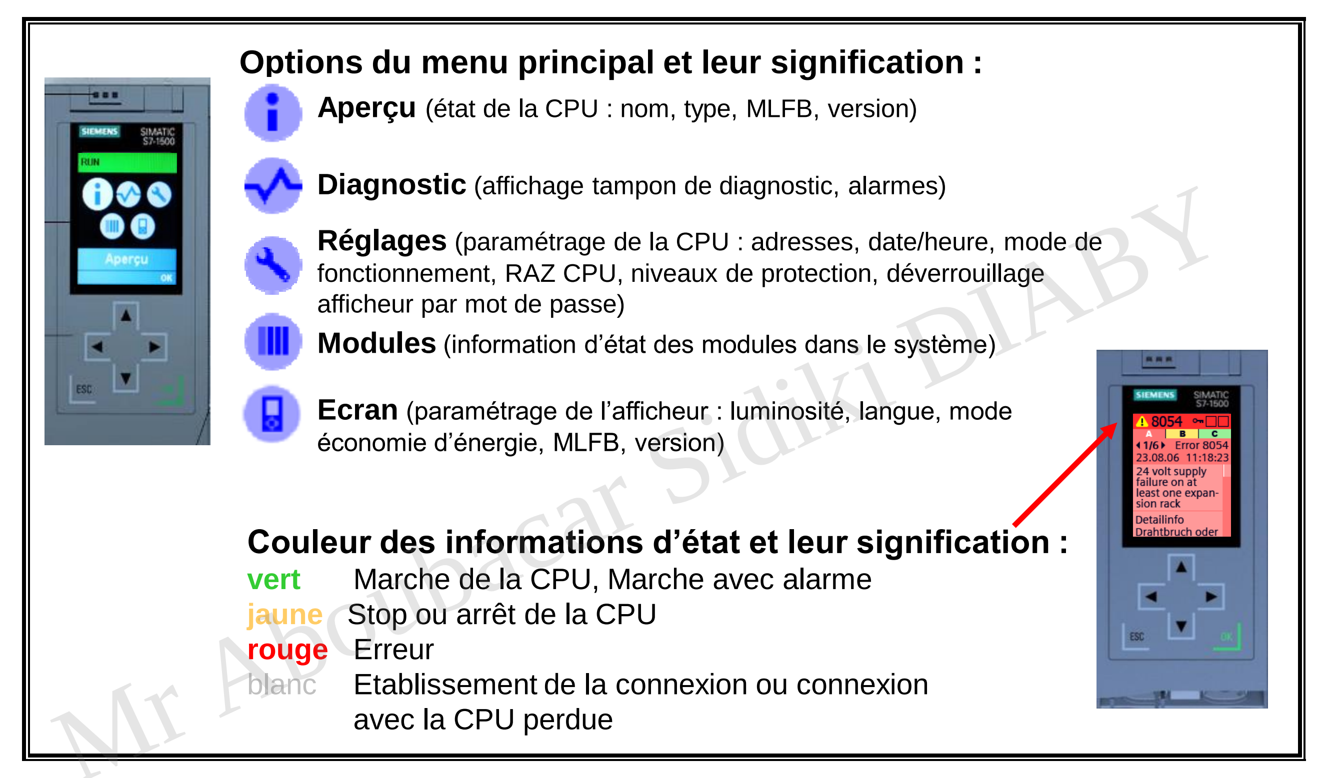
Affichage en deux langues (menus et messages d'erreur/signalisation)

Changement de langue possible en cours de fonctionnement

Utilisation



2.7.2.1. SIMATIC S7-1500 : Afficheur de la CPU → Menus et couleurs



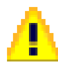
La représentation couleur de l'écran varie selon l'état courant de la CPU.

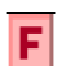
Les langues d'affichage disponibles sont les langues de dialogue disponibles de STEP 7.

Icônes supplémentaires dans l'information d'état

 Le mot de passe est configuré mais non entré

 Le mot de passe est configuré et entré

 Une alarme est présente

 Une tâche de forçage est active dans la CPU

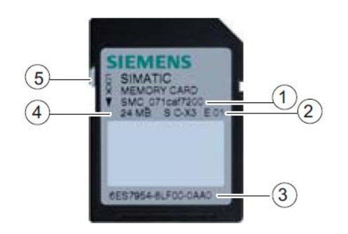
 Sécurité activée. Fonctionnement de sécurité actif (pour CPUs de sécurité)
Si le fonctionnement de sécurité est désactivé l'icône est grisée.

 CPU de sécurité (pour CPUs de sécurité)

Simulateur d'afficheur pour S7-1500

Sous le lien : TIA Portal Information Center > First steps > Getting Started > SIMATIC S7-1500 and STEP7 - Getting Started > Displays of the SIMATIC S7-1500 controller family vous pouvez testez l'afficheur du S7-1500 en version en ligne ou vous pouvez également le télécharger et le tester hors ligne.

2.8. SIMATIC S7-1200/1500 : Cartes mémoire → Vue d'ensemble



① Numéro de série de la carte SD
② Numéro de version
③ Numéro de référence
④ Taille de la carte
⑤ Verrou pour protection écriture (ne doit pas être protégé en écriture)

Ecriture avec :
- Lecteur de carte SD du commerce
- Console de programmation Field-PG M3/M4

Carte mémoire SIMATIC du S7-1200 :
- Extension de la mémoire de chargement
- Transfert de programmes
- Mise à jour du firmware
- Documentation
- Memory Card Binding *(Protection)*
- DB « unlinked »
- Archivage de données
- Remplacement de module sans PG

Carte mémoire SIMATIC du S7-1500 :
- Mémoire de chargement
- Mise à jour du firmware
- Documentation
- Memory Card Binding *(Protection)*
- DB « unlinked »
- Archivage de données
- Remplacement de module sans PG

Memory Card Binding – Protection contre la copie

La possibilité d'exécution du programme peut être liée au numéro de série de la carte.

Mémoire de chargement

- S7-1500
  Le système ne dispose pas de mémoire de chargement intégrée et nécessite impérativement une carte mémoire enfichée.
- S7-1200
  Le système dispose d'une mémoire de chargement intégrée. La carte mémoire enfichée peut remplacer la mémoire de chargement intégrée (extension) ou être utilisée pour la mise à jour de programmes (transfert de programmes).

Transfert de programmes ← uniquement S7-1200

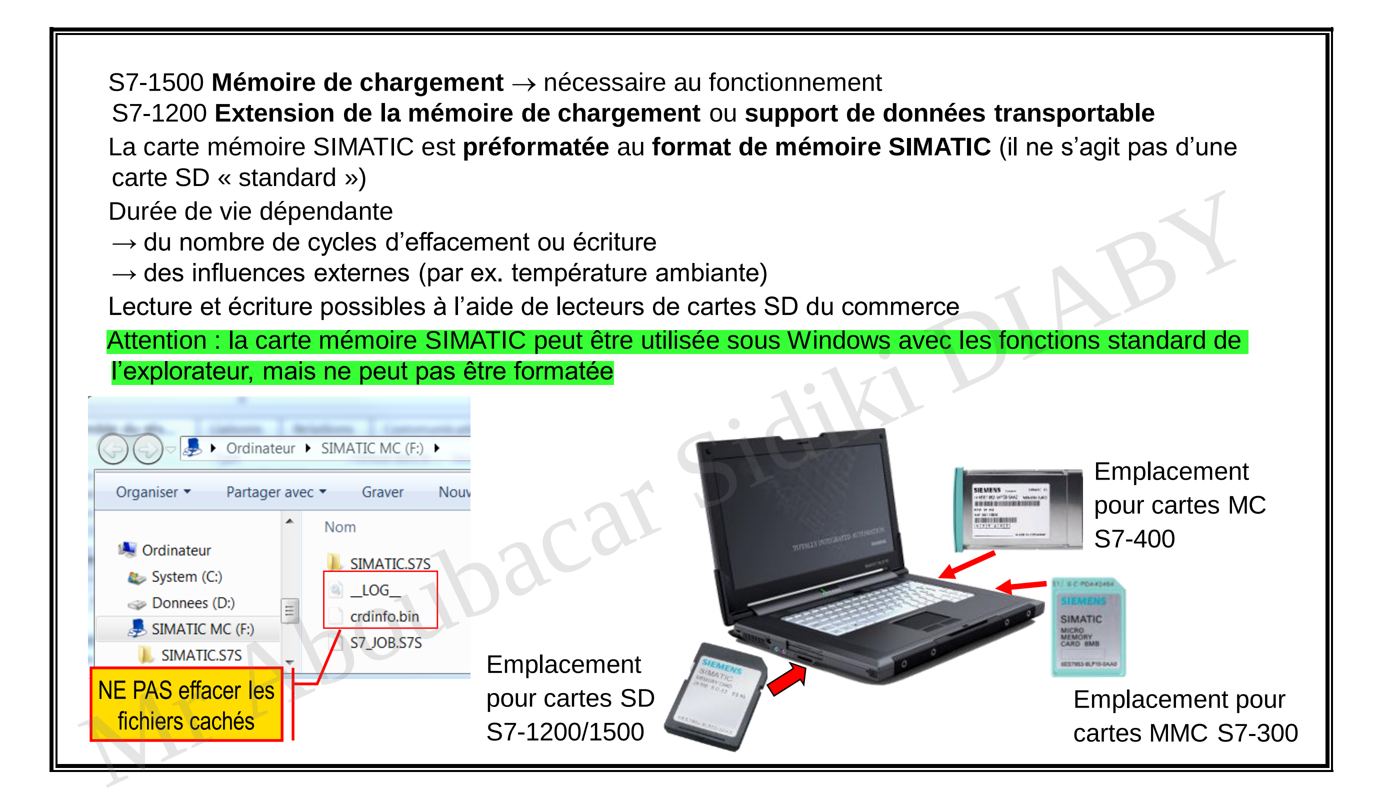
Le transfert de programmes via la carte (mode « transfert ») n'est possible que pour le S7-1200. L'enfichage d'une carte permet dans ce cas de charger un programme dans la CPU sans recourir à une console de programmation (PG).

Archivage de données ← uniquement S7-1500

Il est possible d'archiver des valeurs de process sur la carte.


L'utilisation de cette fonctionnalité a une influence sur la durée de vie de la carte mémoire

2.8.1. SIMATIC S7-1200/1500 : Cartes mémoire → Caractéristiques



Sens d'insertion des cartes SD ou MMC

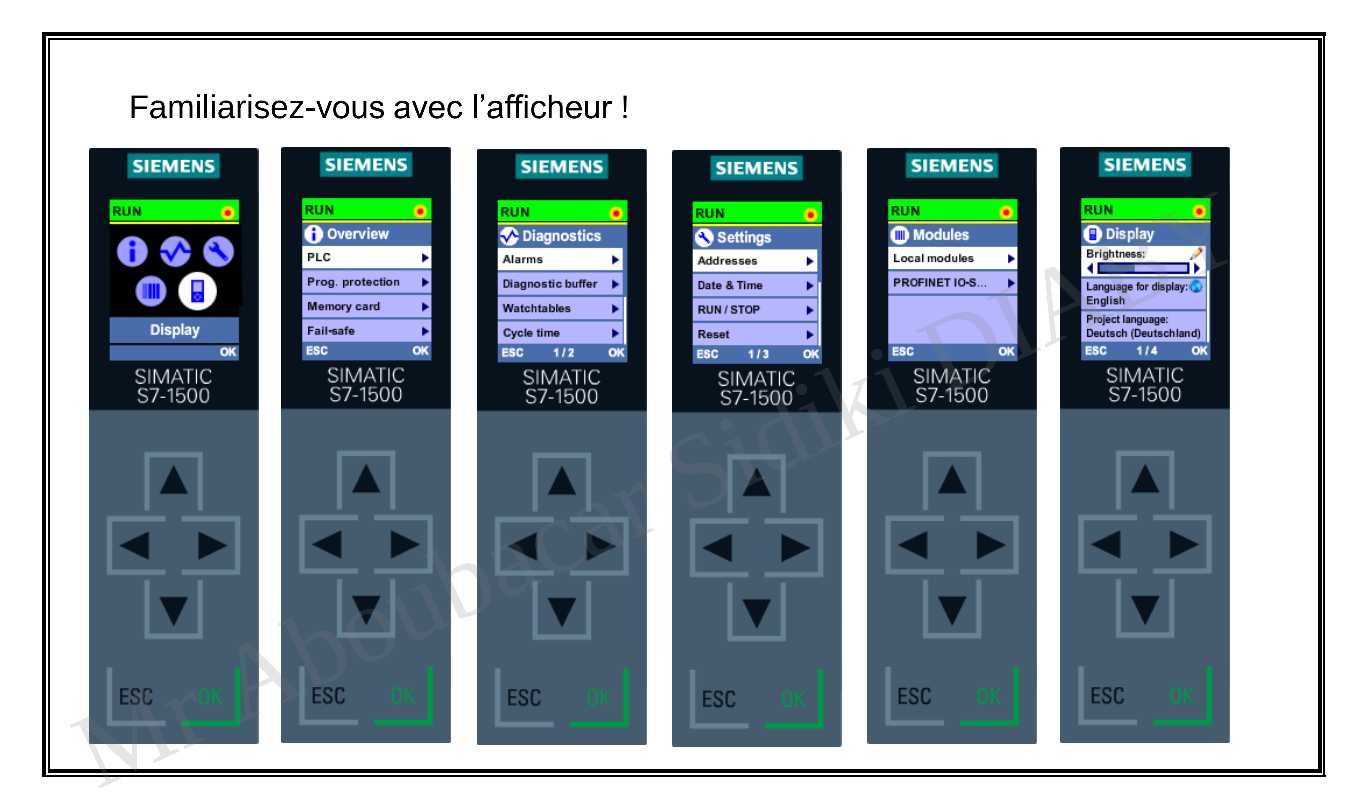
Avec contacts vers le haut

Nombre de cycles d'effacement/écriture

La carte mémoire SIMATIC autorise au minimum 500 000 cycles d'effacement/écriture à une température ambiante n'excédant pas 60 °C.

2.9. Exercice 1: Afficheur

Familiarisez-vous avec l'afficheur !



Enoncé:

1. Changez la langue et la luminosité de l'afficheur.
2. Vérifiez le type de carte mémoire et la capacité mémoire restante.
3. Visualisez le tampon de diagnostic.
4. Notez le MLFB (numéro d'article) et la version firmware de la carte analogique centralisée
5. Changez l'heure, l'adresse IP de l'interface et formater la carte mémoire.

Attention:

Il sera peut être impossible de formater la carte. C'est le cas lorsque l'accès en écriture de l'afficheur a été désactivé pour le dernier projet chargé. (Réglage de la CPU)

Dans ce cas la carte est uniquement effaçable sous Windows ou dans TIA Portal.

2.9.1. Exercice 2: Charger le programme sur la mémoire

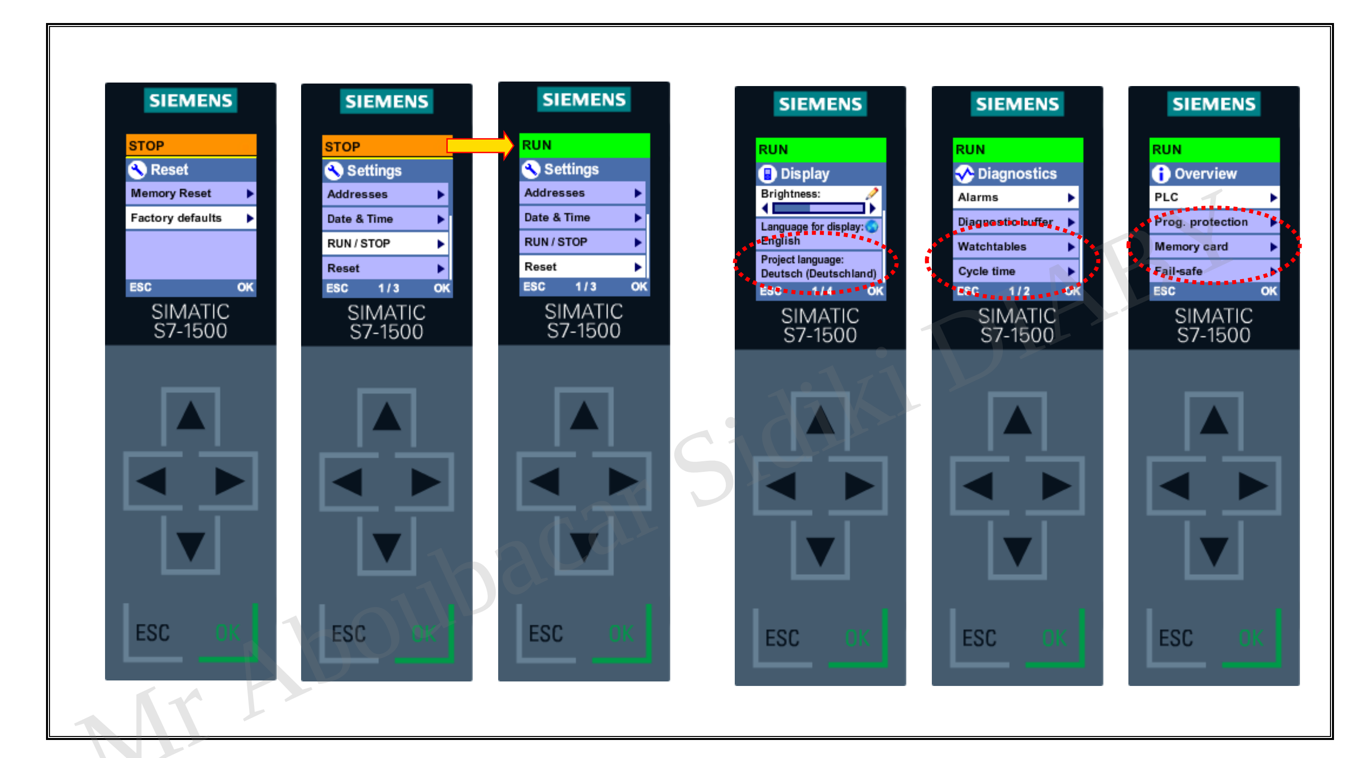
Enoncé:

1. Mettre la CPU en STOP à l'aide de l'afficheur.
2. Retirez la carte mémoire SIMATIC.
3. Insérez la SMC dans le lecteur de carte SD de votre console de programmation (PC).
4. Effacez la carte sous Windows Explorer.

Attention: N'effacez pas les fichiers cachés!!!

5. Ouvrez le répertoire « D:\Exercices_TIA-Portal\TIA-SYSUP\SMC-Project ».
6. Désarchivez le fichier ZIP (nom en accord avec le MLFB et le firmware de la CPU) vers la carte mémoire SIMATIC.

2.9.2. Exercice 3: Diagnostic et test du programme



Enoncé:

1. Insérez la carte mémoire SIMATIC Memory dans la CPU et attendez la fin du clignotement de la Del MAIN RUN STOP.
2. Effectuez un reset de la CPU aux réglages d'usine en utilisant l'afficheur (pour effacer les données de l'ancien programme qui pourrait encore se trouvez dans la CPU)
3. Passez la CPU en Run.
4. Choisissez la langue.
5. Passez aux diagnostics en utilisant l'afficheur: vérifiez les tables de visualisation, le temps de cycle et la mémoire.
6. Vérifiez que le câble de votre convoyeur soit connecté sur l'embase « S7-1500 DI/DO » au dos de votre maquette de formation.
7. Alimentez votre maquette de convoyeur.
8. Effectuez un test de programme.

   Description de la fonction

   Le convoyeur sert à transporter des pièces, il peut être utilisé suivant deux modes d'exploitation. (Mode manuel ou Automatique)
   Cette sélection peut être effectuée en utilisant les boutons « En Service » et « Hors Service ».

   Mode Manuel « Q En service » = désactivé

   Le moteur du convoyeur peut être piloté par impulsions vers la droite et vers la gauche.

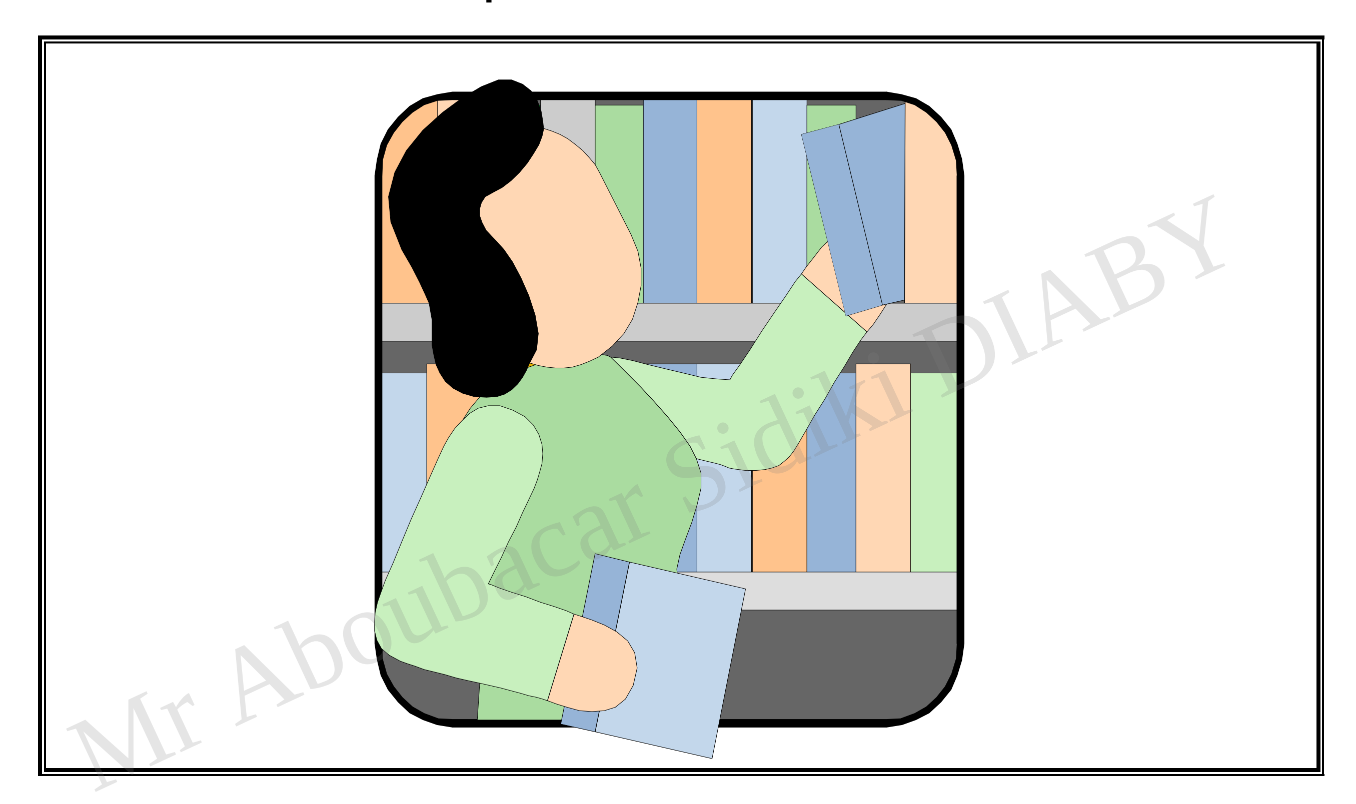
   Mode Automatique « Q En service » = activé

   Le convoyeur transporte les pièces des postes 1,2 ou 3 vers la droite jusqu'à la barrière lumineuse, en utilisant le bouton attribué à chaque poste. Un seul emplacement est occupé à la fois.
   La signalisation aux postes 1, 2 et 3 se caractérise par...

   – une signalisation allumée en continu sur le poste sur lequel est détectée une pièce

   – un clignotement à 2Hz aussi longtemps que le moteur de la bande transporteuse tourne.

2.10. Informations complémentaires





Mr Aboubacar Sidiki DIABY

2.10.1. Objets technologiques - Régulation PID

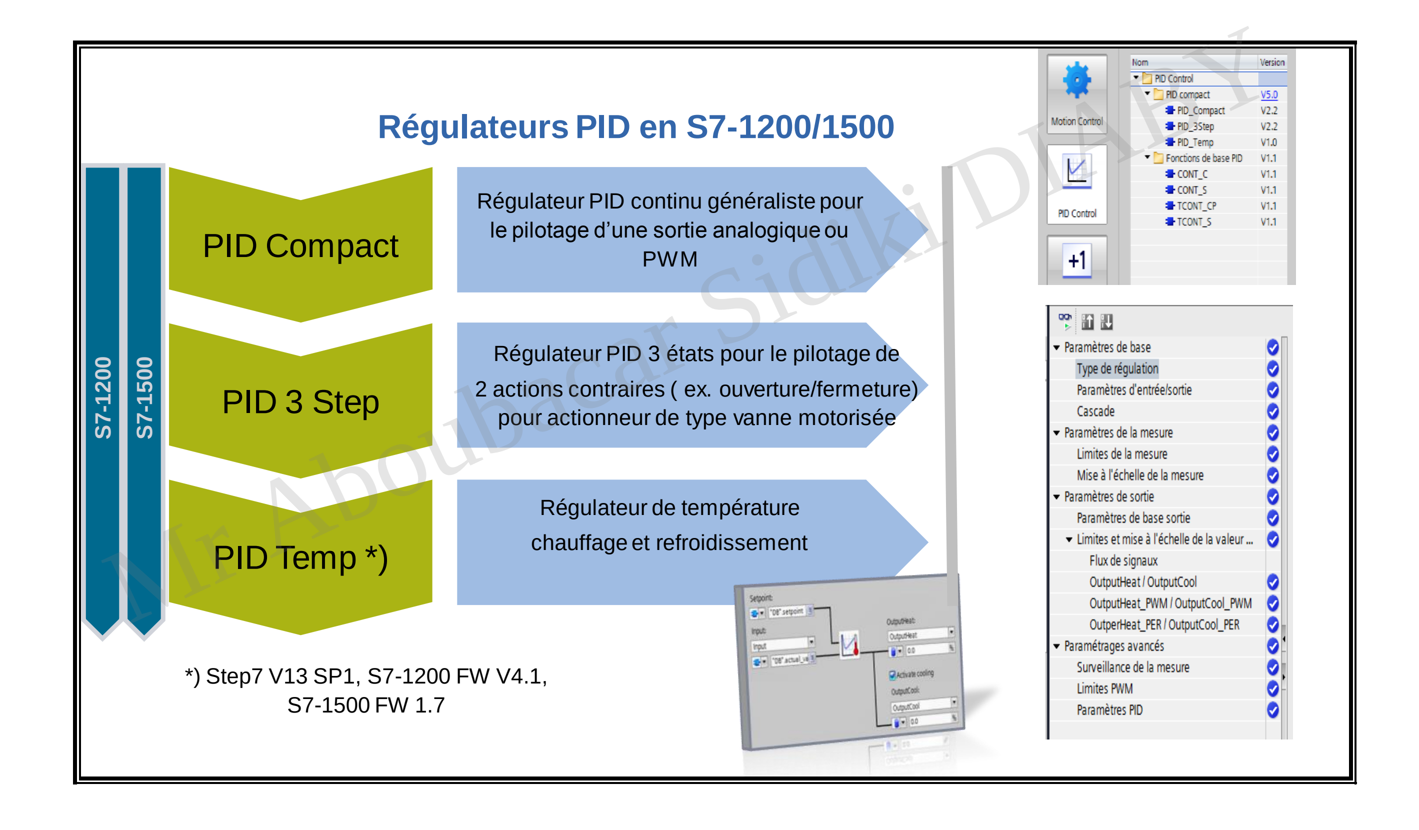
Régulation PID intégrée

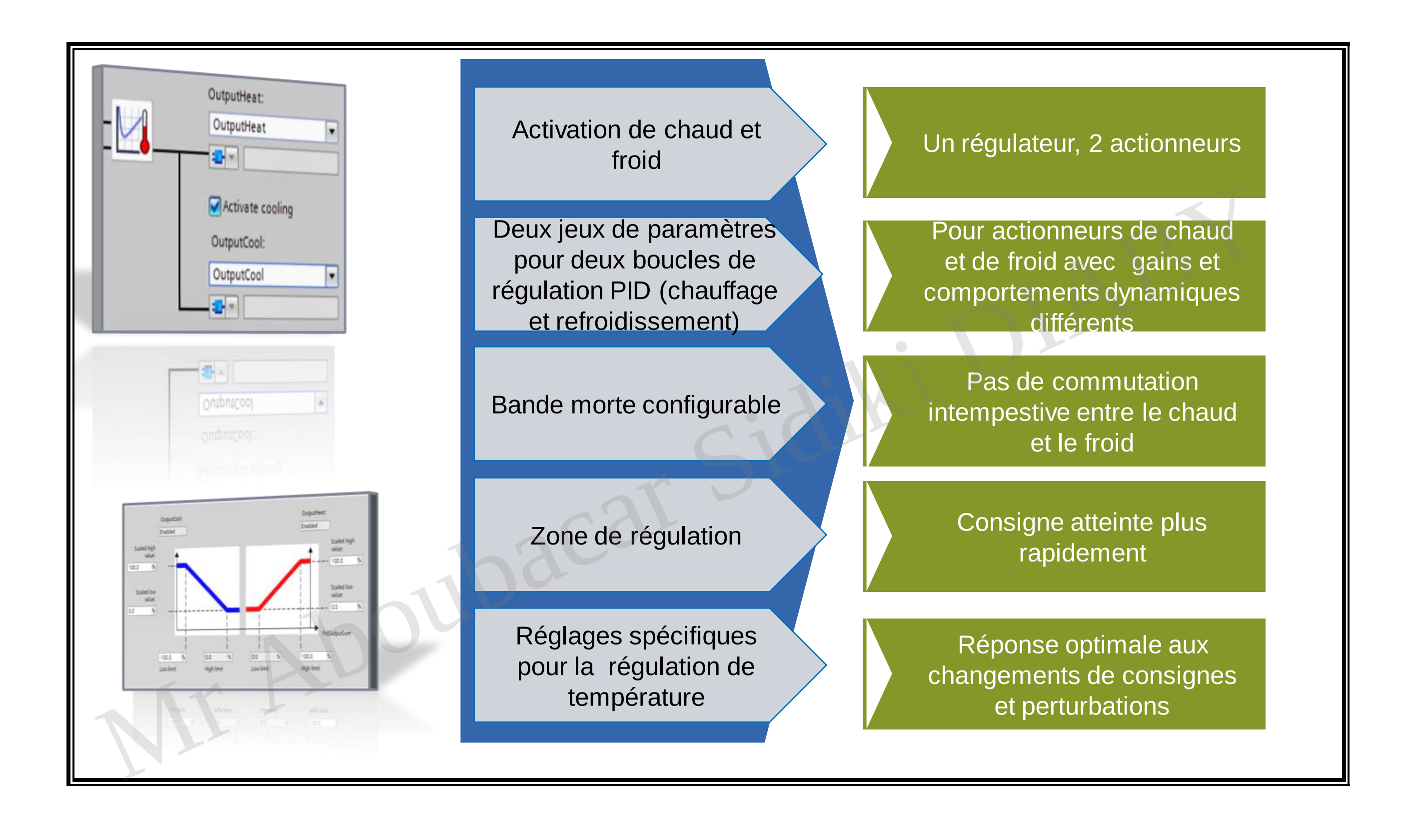
- ❑ Pour les tâches de régulation simples
- ❑ Réalisation d'une modulation de largeur d'impulsion continue et discrète ou de régulateurs pas-à-pas
- ❑ Mise en service facile grâce au réglage automatique
- ❑ Multiples mécanismes de protection des personnes et des machines

PID Compact/PID 3-Step

- ❑ Régulateur continu à la sortie analogique ou à modulation de largeur d'impulsions ou régulateur pas-à-pas spécialisé pour actionneurs intégrés (p. ex. vannes)
- ❑ Personnalisation (auto optimisation)

Réduction des coûts grâce aux régulateurs PID intégrés à la personnalisation haut de gamme



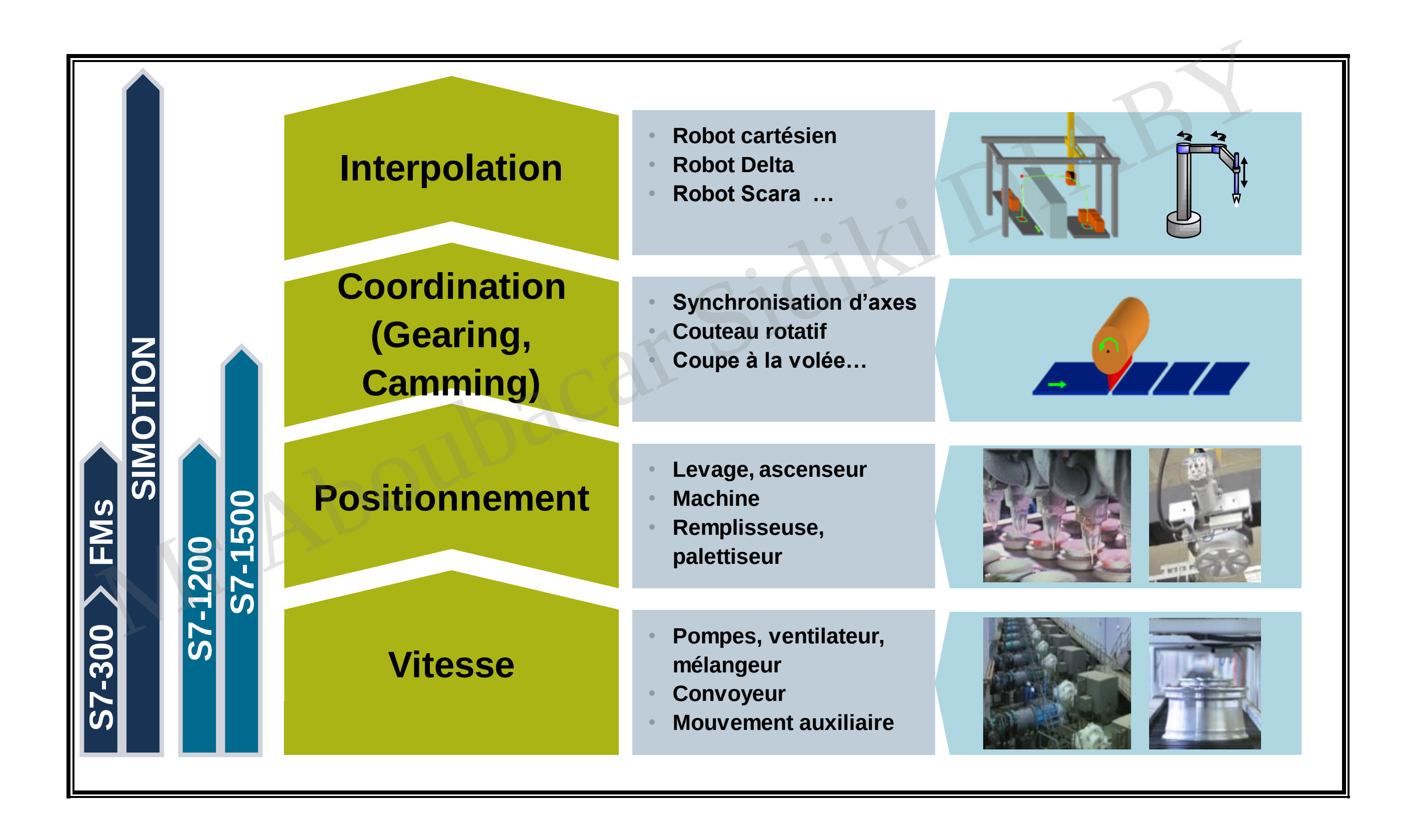

OutputHeat:
OutputHeat
Activate cooling
OutputCool:
OutputCool
Activation de chaud et froid
Deux jeux de paramètres pour deux boucles de régulation PID (chauffage et refroidissement)
Bande morte configurable
Zone de régulation
Réglages spécifiques pour la régulation de température
Un régulateur, 2 actionneurs
Pour actionneurs de chaud et de froid avec gains et comportements dynamiques différents
Pas de commutation intempestive entre le chaud et le froid
Consigne atteinte plus rapidement
Réponse optimale aux changements de consignes et perturbations

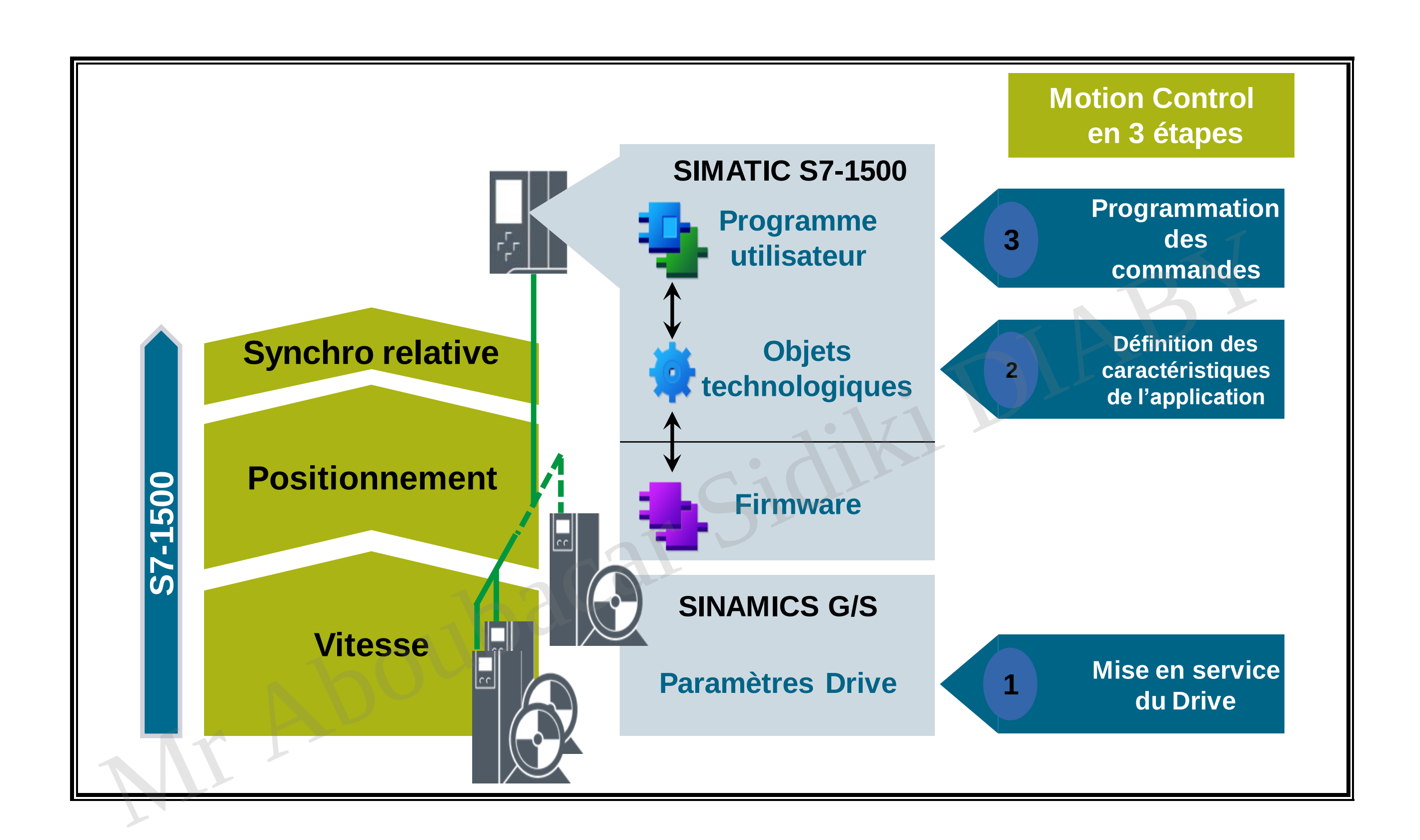
2.10.2. Objets technologiques – Motion Control sur S7 1200 et S7 1500

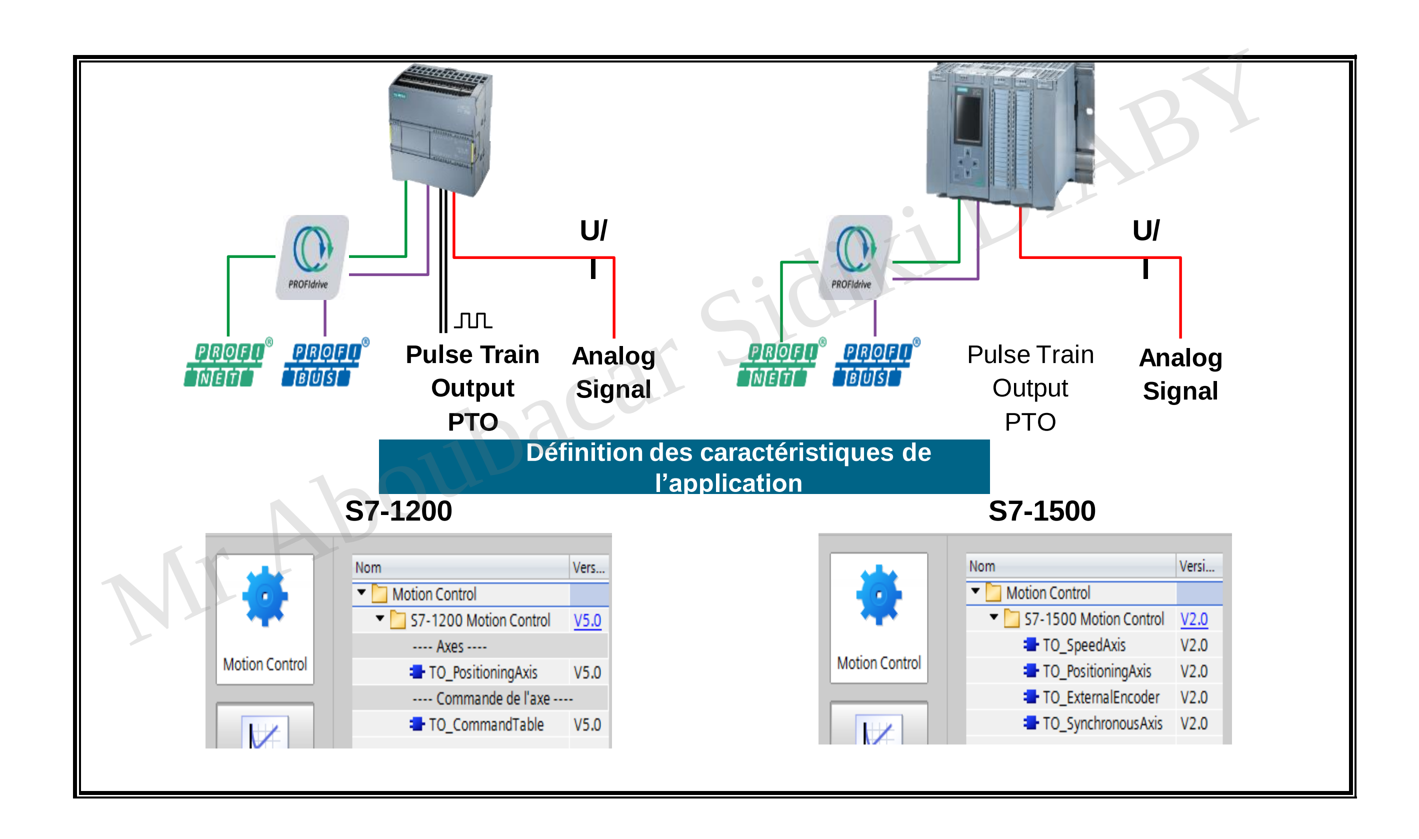
Fonction Motion Control

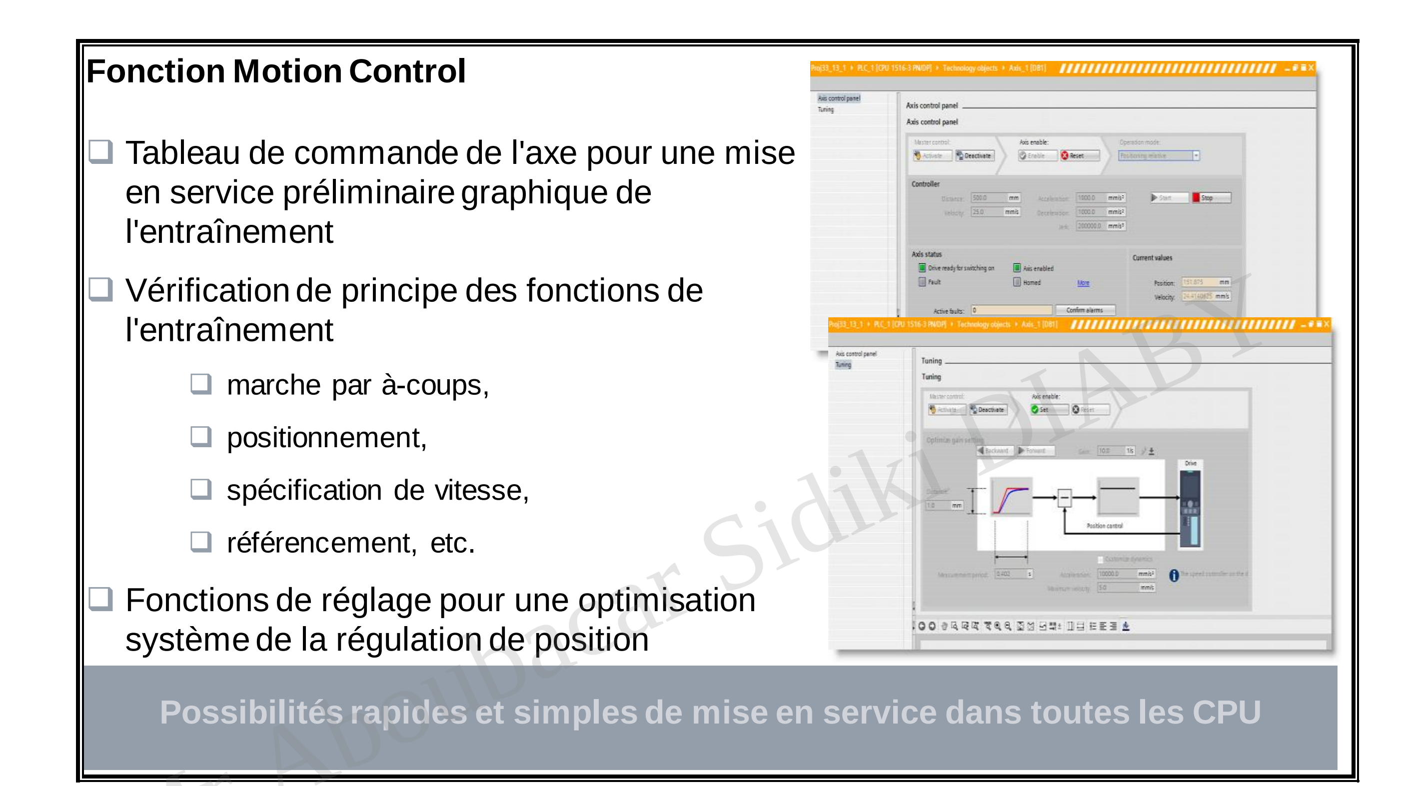
- Connecter, paramètrer, programmer et mettre en service un variateur de vitesse (compatible Profidrive) avec définition de l'aplication
- Programmation simple de déplacements avec les blocs
  PLCopen Motion (Instructions)
- Outils conviviaux de diagnostic et de mise en service
  (pupitre de commande, fonction Trace)
- Recherche simplifiée d'erreurs par des messages d'alarme automatiques au système d'ingénierie et à l'IHM
- Applicatifs : Axe de vitesse, axe de positionnement

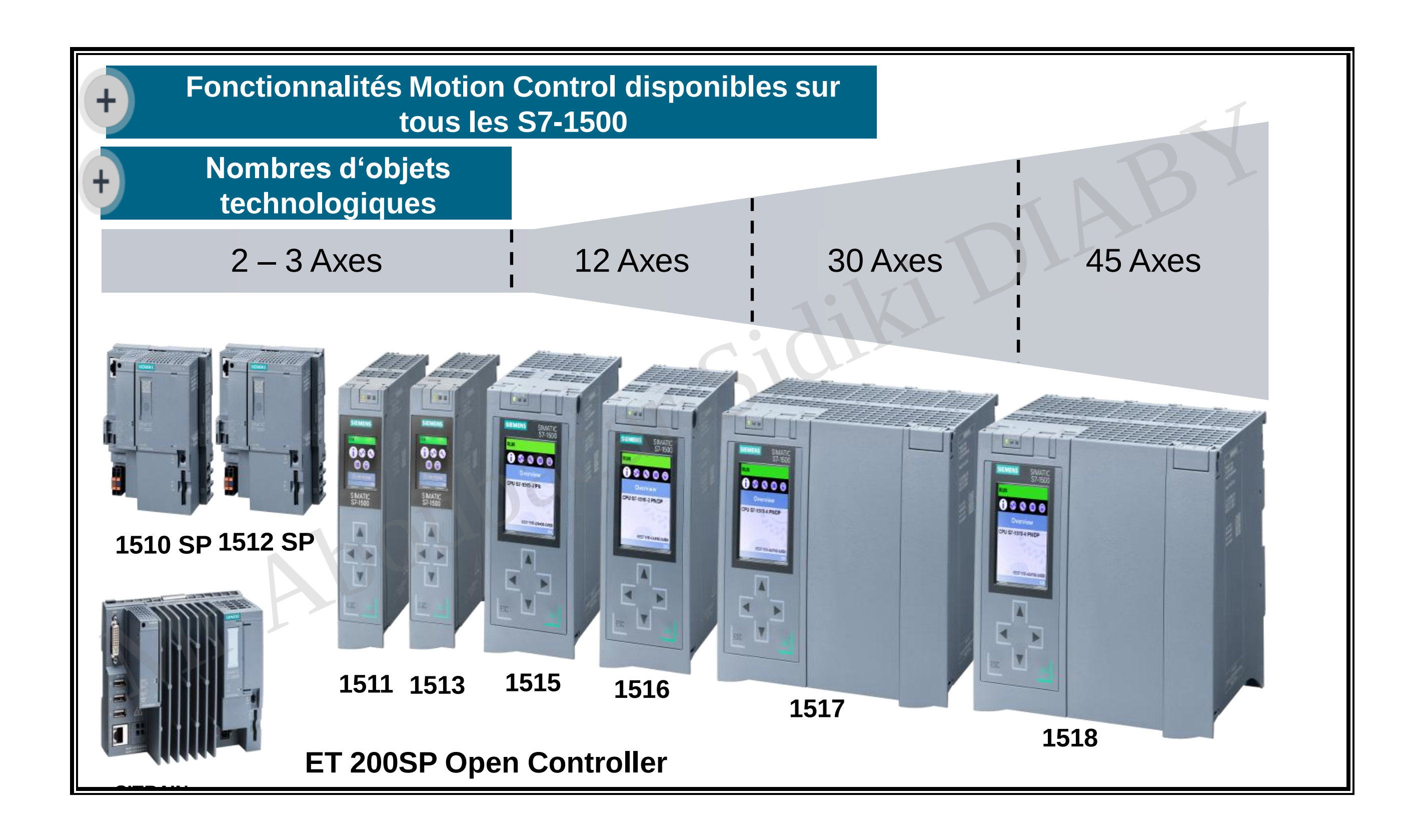
Insérer un drive de façon standard dans un projet d'automatisme



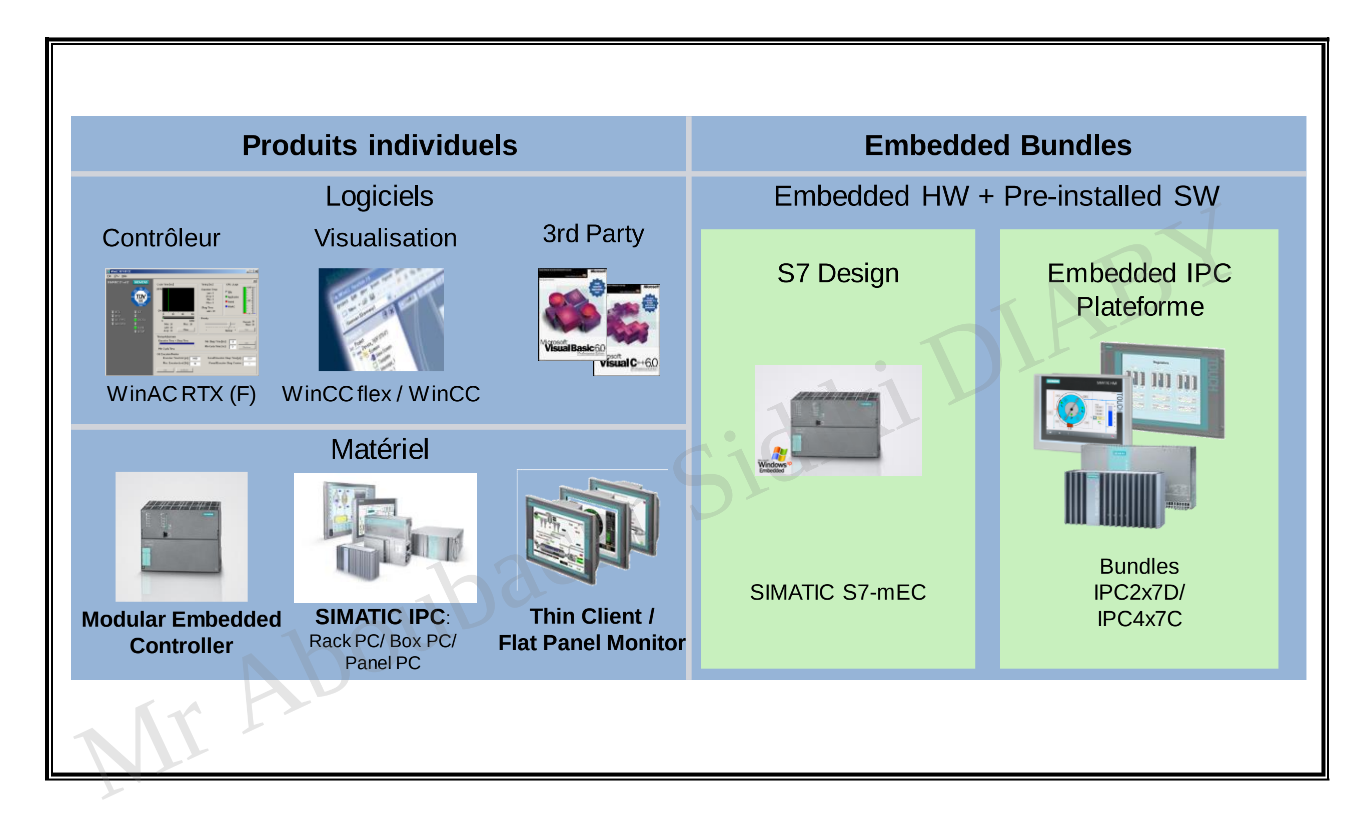

Motion Control
en 3 étapes
SIMATIC S7-1500
Programme
utilisateur
Objets
technologiques
Firmware
SINAMICS G/S
Paramètres Drive
Synchro relative
Positionnement
Vitesse
S7-1500
3
Programmation
des
commandes
2
Définition des
caractéristiques
de l'application
1
Mise en service
du Drive


U/
I
Pulse Train
Output
PTO
Analog
Signal
U/
I
Pulse Train
Output
PTO
Analog
Signal
Définition des caractéristiques de
l'application
S7-1200
S7-1500
Motion Control
Nom
Vers...
Motion Control
S7-1200 Motion Control
V5.0
---- Axes ----
TO_PositioningAxis
V5.0
---- Commande de l'axe ----
TO_CommandTable
V5.0
Motion Control
Nom
Versi...
Motion Control
S7-1500 Motion Control
V2.0
TO_SpeedAxis
V2.0
TO_PositioningAxis
V2.0
TO_ExternalEncoder
V2.0
TO_SynchronousAxis
V2.0


Fonction Motion Control
Tableau de commande de l'axe pour une mise en service préliminaire graphique de l'entraînement
Vérification de principe des fonctions de l'entraînement
marche par à-coups,
positionnement,
spécification de vitesse,
référencement, etc.
Fonctions de réglage pour une optimisation système de la régulation de position
Possibilités rapides et simples de mise en service dans toutes les CPU


Fonctionnalités Motion Control disponibles sur tous les S7-1500
Nombres d'objets technologiques
2 – 3 Axes
12 Axes
30 Axes
45 Axes
1510 SP
1512 SP
1511
1513
1515
1516
1517
1518
ET 200SP Open Controller

2.10.3. SIMATIC PC based Automation

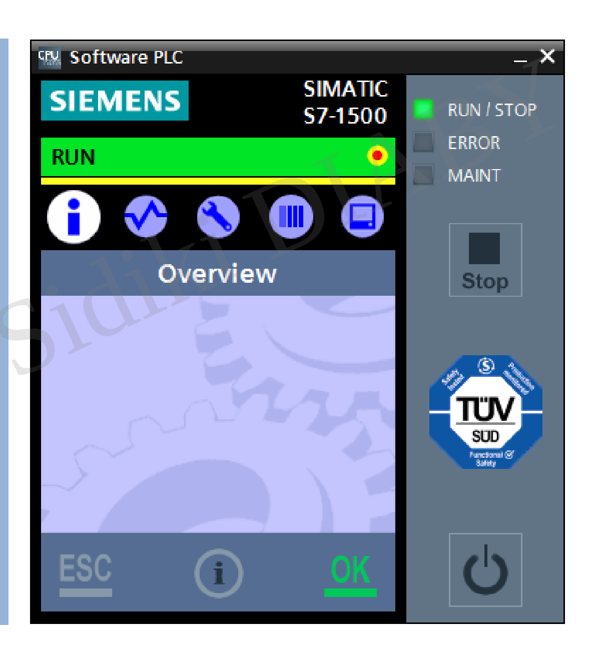


Présentation de la gamme

La figure ci-dessus présente les principales composantes de l'automatisation sur PC.

2.10.1. Contrôleur Software

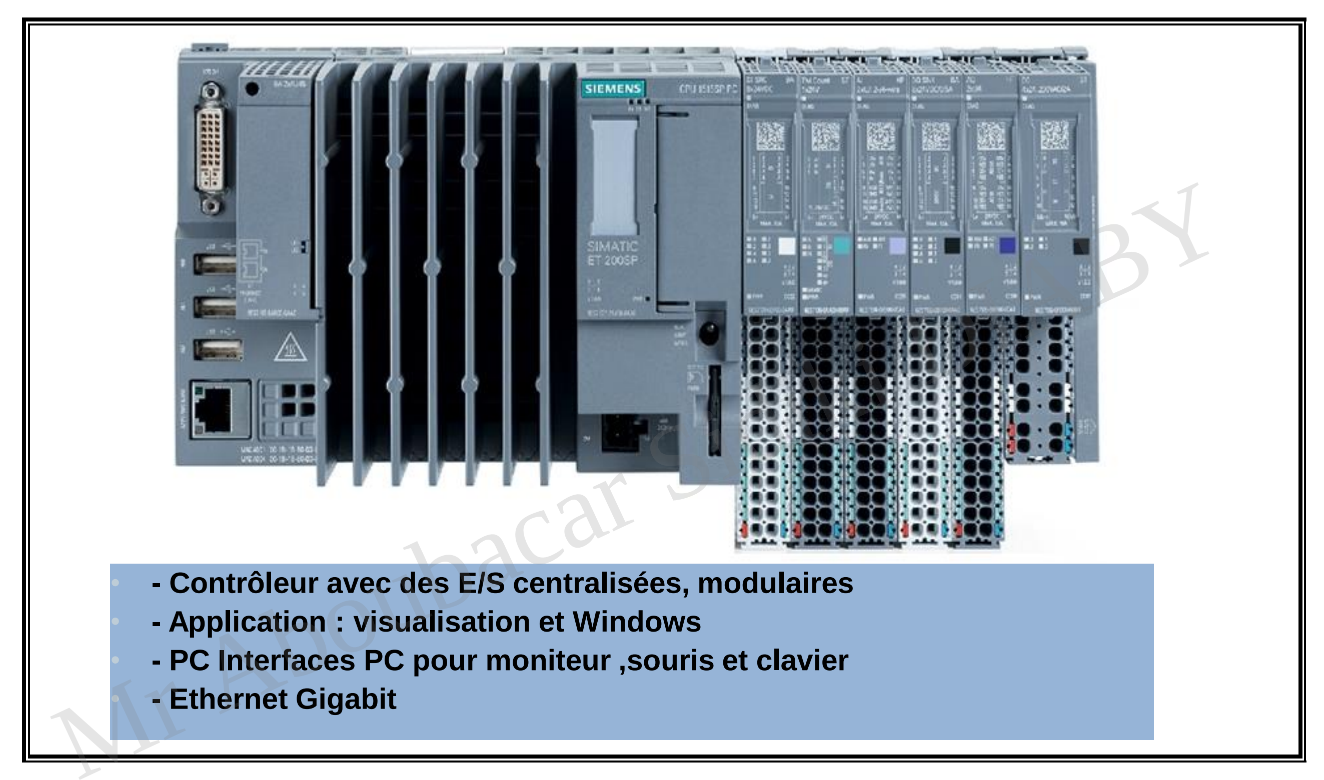
- **Utilisation sur PC pour l'industrie : SIMATIC IPCs**
- **Opérationnel en totale indépendance du système Windows (même en cas de redémarrage ou problème avec le système)**
- **Contrôleur flexible pour exploitation de machines spéciales avec un haut niveau de performances et de technicité**
- **Intégration de fonctions spécifiques à l'utilisation via des interfaces (par ex. pour C++ / Matlab)**



Plus d'informations disponibles en suivant le lien :

TIA Portal Information Center → Informations produits → Contrôleurs logiciels

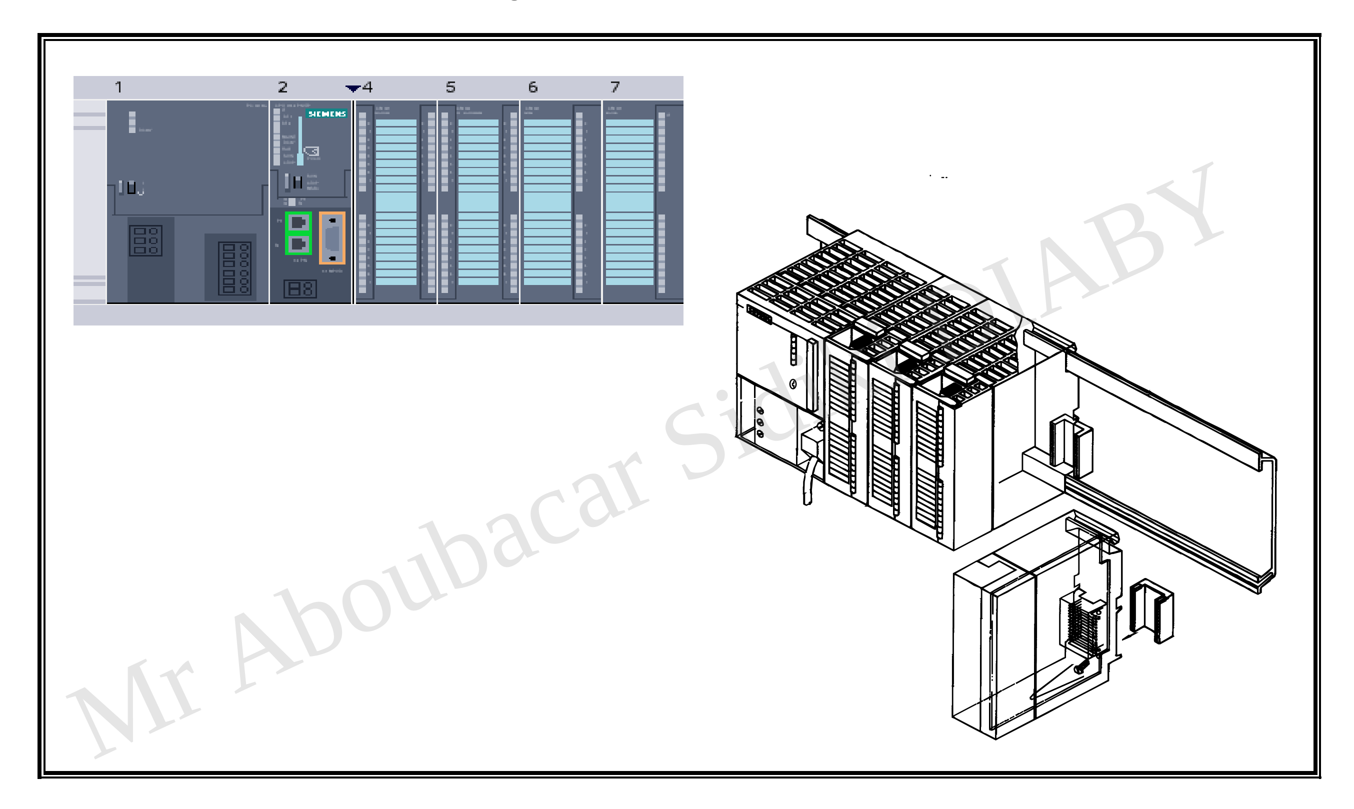
2.10.2. ET200SP Open Controller « tout en un »



Plus d’informations disponibles en suivant le lien :

TIA Portal Information Center → Informations produits → Contrôleurs distribués

2.11. SIMATIC S7-300 : Système d'automatisation modulaire

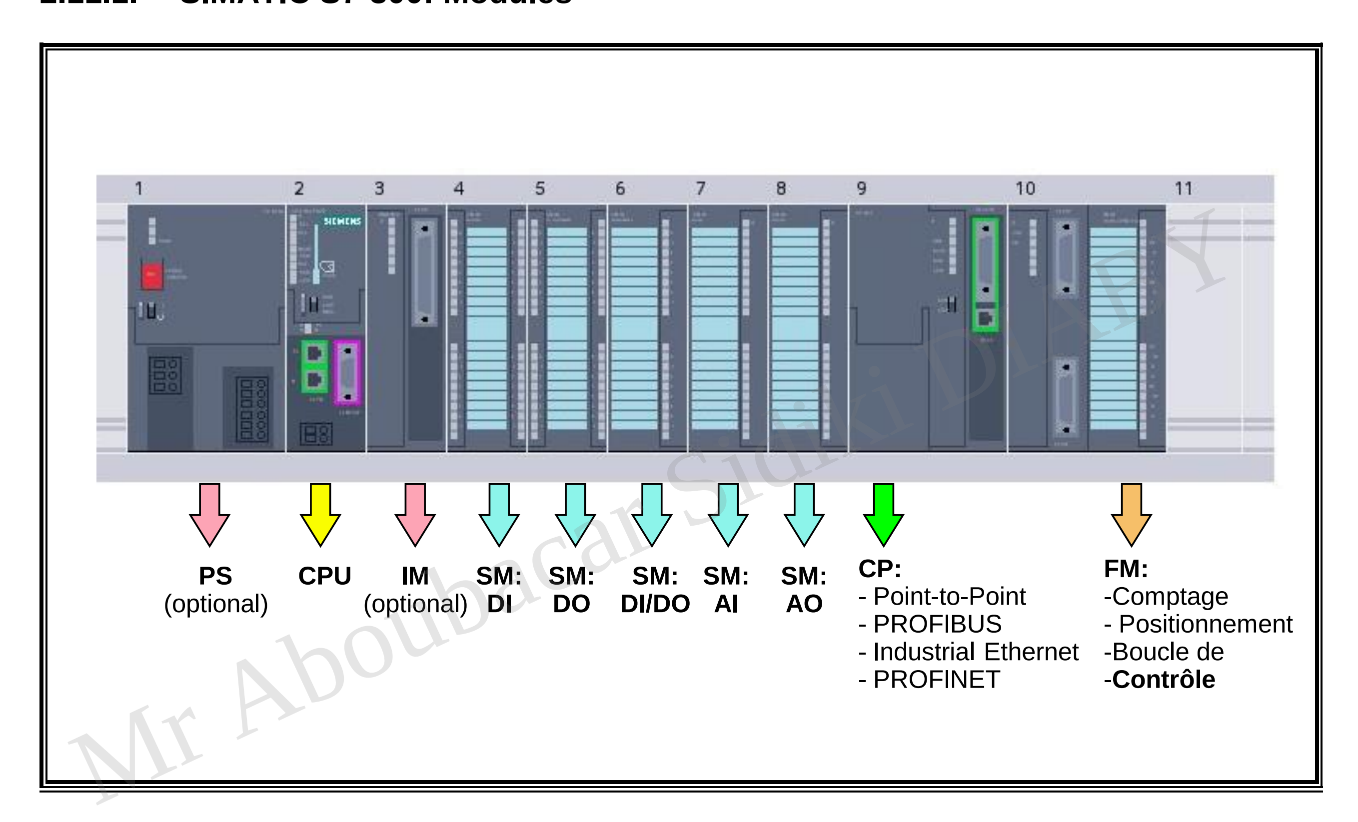


Caractéristiques

Mini-automate modulaire pour les applications d'entrée et de milieu de gamme

- Gamme échelonnée de CPU
- Gamme complète de modules
- Possibilité d'extension jusqu'à 32 modules
- Bus de fond de panier intégré aux modules
- Possibilité de mise en réseau avec
  - l'interface multipoint (MPI),
  - PROFIBUS ou
  - Industrial Ethernet ou
  - PROFINET
- Raccordement central de la PG (console de programmation) avec accès à tous les modules
- Liberté de montage aux différents emplacements
- Configuration et paramétrage à l'aide de l'outil « Configuration matérielle »

2.11.1. SIMATIC S7-300: Modules



Modules de signaux (SM)

Modules ETOR:	24V=, 120/230V~
Modules STOR :	24V=, relais
Modules EANA :	tension, courant, résistance, thermocouple
Modules SANA :	tension, courant

Coupleurs (IM)

Les coupleurs IM360/IM361 ou IM365 permettent de réaliser des configurations matérielles multi châssis. Le bus est relié en boucle entre les différents châssis.

Modules de réservation (DM)

Le module de réservation DM 370 occupe un emplacement pour un module de signaux non paramétré. Cet emplacement est donc réservé, par exemple pour le montage ultérieur d'un coupleur.

Modules de fonction (FM)

Comptage

Positionnement

Régulation

Modules de communication (CP)

Liaison point-à-point

PROFIBUS

Industrial Ethernet

PROFINET

2.12. SIMATIC S7-400 : Système d'automatisation modulaire

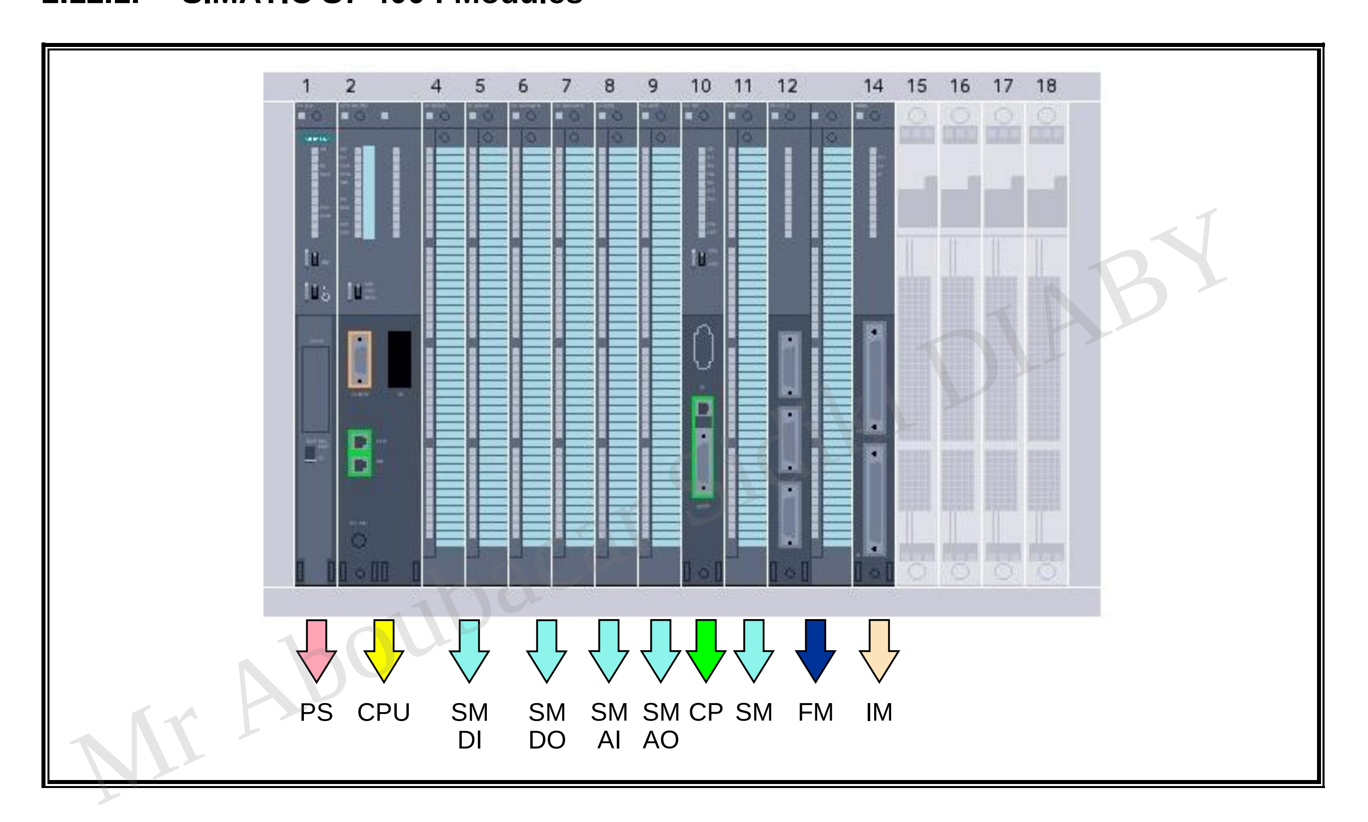


Caractéristiques

Automate haute performance pour les applications de milieu et haut de gamme,

- Gamme échelonnée de CPU,
- Gamme complète de modules,
- Possibilité d'extension à plus de 300 modules,
- Bus de fond de panier intégré au châssis,
- Possibilité de mise en réseau avec
  - l'interface multipoint (MPI),
  - PROFIBUS ou
  - Industrial Ethernet, ou
  - PROFINET,
- Raccordement central de la console de programmation (PG) avec accès à tous les modules,
- Liberté de montage aux différents emplacements,
- Configuration et paramétrage à l'aide de l'outil « Configuration matérielle »
- Mode multiprocesseur (4 CPU maxi. peuvent être mises en œuvre sur un châssis central)

2.12.1. SIMATIC S7-400 : Modules



Modules de signaux (SM)

Modules ETOR : 24 V=, 120/230V~

Modules STOR : 24 V=, Relais

Modules EANA : tension, courant, résistance, thermocouple

Modules SANA : tension, courant

Coupleurs (IM)

Les coupleurs IM460, IM461, IM463, IM467 assurent la liaison entre les différents châssis :

Châssis universel UR1 (Universal Rack) d'une capacité maxi. de 18 modules

Châssis universel UR2 (Universal Rack) d'une capacité maxi. de 9 modules

Châssis universel ER1 (Extension Rack) d'une capacité maxi. de 18 modules

Châssis universel ER2 (Extension Rack) d'une capacité maxi. de 9 modules.

Modules de fonction (FM)

Comptage

Positionnement

Régulation

Processeurs de communication (CP)

Couplage point-à-point

PROFIBUS

Industrial Ethernet

PROFINET

Sommaire 3

3. Logiciel d’ingénierie

A l’issue du chapitre, vous allez ...

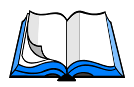


... avoir un aperçu des fonctionnalités de la nouvelle plateforme d’ingénierie

... connaître les produits et les systèmes d’ingénierie

... connaître l’interface utilisateur de la nouvelle plateforme

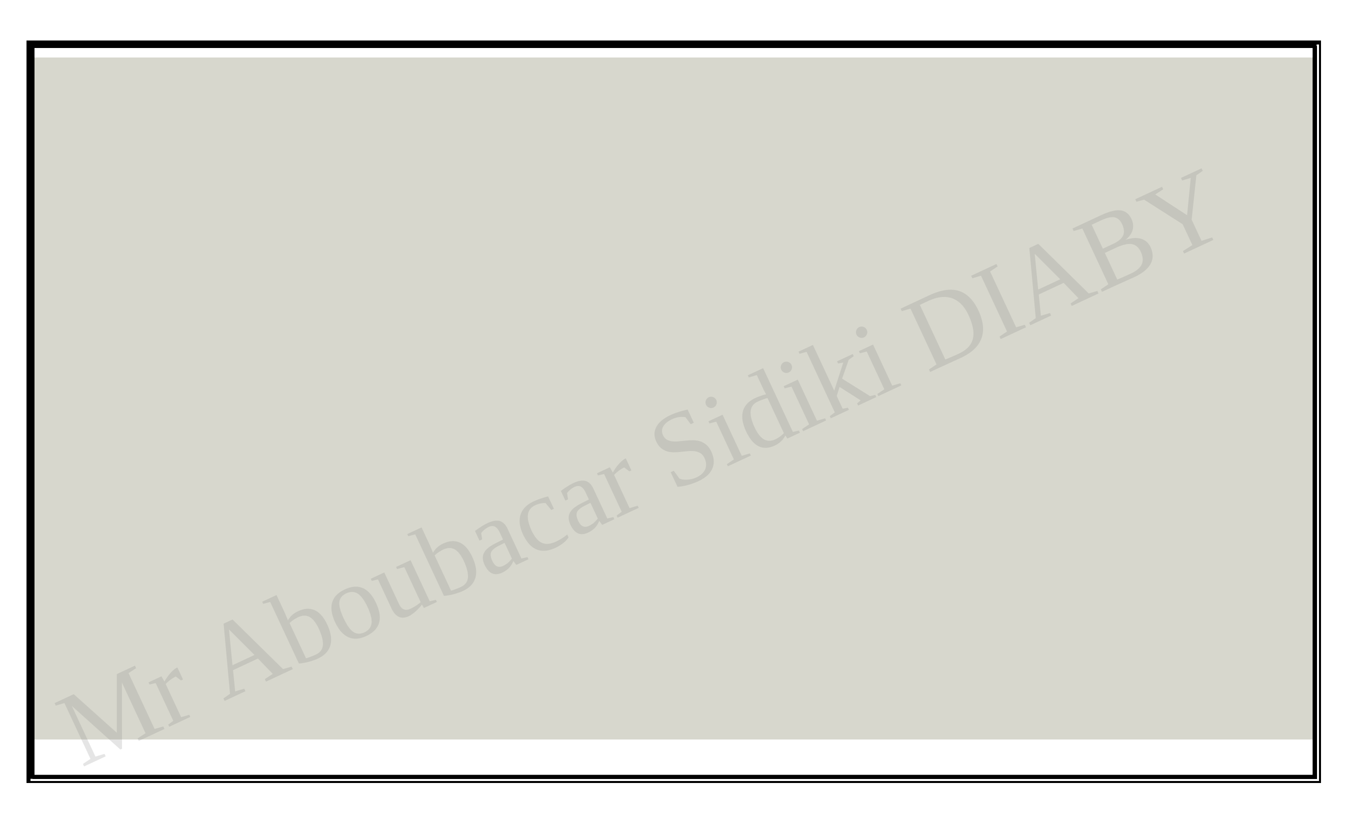
... Avoir un aperçu des possiblités de téléchargement des programmes en ligne



Ce chapitre présente les principales composantes de la plateforme TIA Portal.

A l’issue de ce chapitre, le participant au stage connaîtra l’offre logicielle STEP7 et WinCC.

3.1. Gestion du cycle de vie des produits



Cycle de vie des produits

Les produits et processus deviennent de plus en plus complexes. Siemens fournit des solutions de gestion du cycle de vie d'un produit (PLM) pour toutes les étapes de l'avant-projet à la fin de vie du produit ce qui permet de réussir le développement, la fabrication et la mise sur le marché des produits.

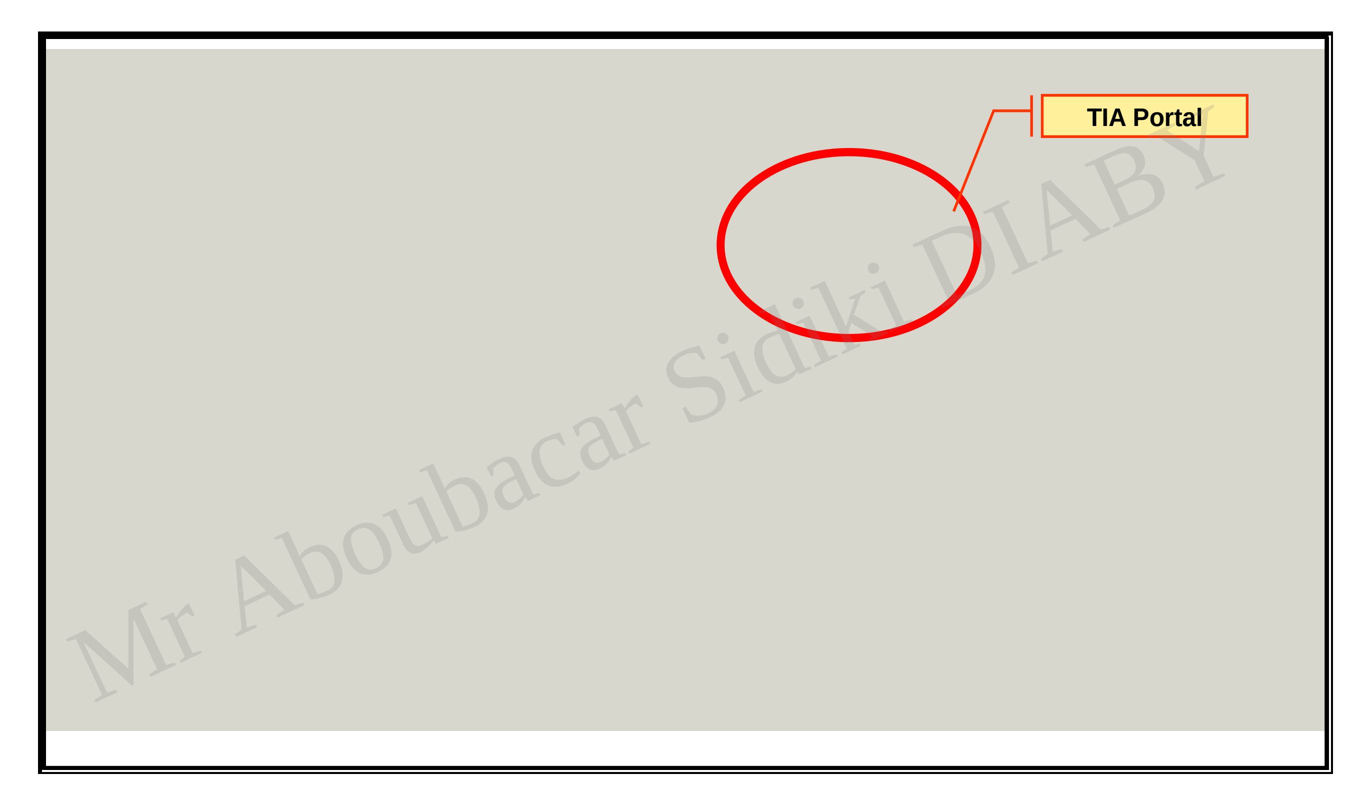
L'approche globale Siemens consiste à transformer la chaîne de valeur ajoutée traditionnelle en un cycle de vie du produit et de sa fabrication. Elle englobe la conception du produit, la planification de la production, la technologie de fabrication jusqu'à la production et les prestations de services.

Seul un modèle d'entreprise complètement numérique peut offrir les ressources et la flexibilité pour accélérer les processus et optimiser les processus de production.

Cela implique également un système unique de gestion et de stockage des données. Avec « Teamcenter », Siemens propose la plateforme industrielle innovante pour relier toutes les étapes de la chaîne de valeur ajoutée, la base d'intégration des données multi domaine.

Dans cette chaîne holistique de valeur ajoutée se trouve la plateforme de Cloud « MindSphere » écosystème ouvert pout l'internet des objets.

3.2. Suite logicielle Digital Enterprise -> la réponse pour l'industrie 4.0



TIA Portal – votre portail pour l'automatisation dans l'entreprise numérique

Totally Integrated Automation Portal (TIA Portal) vous fournit un accès illimité à l'ensemble de vos services numériques d'automatisation, de la planification numérique et de l'ingénierie intégrée à la transparence du processus.

La nouvelle version réduira votre temps de mise sur le marché avec, par exemple, l'utilisation d'outil de simulation, augmentera la productivité de votre entreprise avec des fonctions de diagnostic étendu et gestion de l'énergie, et permettra une grande flexibilité de connexion vers le niveau du management. Les nouvelles options avantagent les intégrateurs, les constructeurs de machines ainsi que les opérateurs.

TIA Portal représente la passerelle idéale de l'automatisation dans l'entreprise digitale. En tant que partie de la suite logicielle « Digital Enterprise » avec PLM (Product LifeCycle Management) et MES (Manufacturing Execution System), elle complète l'offre globale Siemens dans l'industrie pour son passage vers l'industrie 4.0.

3.3. TIA Portal – Plateforme d'ingénierie pour solutions d'automatisation entièrement intégrées

Totally Integrated Automation Portal est un **environnement d'ingénierie** qui est le cadre d'une ingénierie cohérente pour…

- …La programmation des systèmes automatisés
- …La visualisation des processus

Siemens Totally Integrated Automation Portal

Totally Integrated Automation Portal

Start

Devices & Networks

PLC Programming

Visualization

Online & Diagnostics

STEP 7

Safety

WinCC

Startdrive

SIMATIC Controller

Safety

SIMATIC HMI

SINAMICS

…

Les progiciels qui tournent en mode autonome sont souvent limités en termes d'interopérabilité et d'intégration.

Seul un environnement de travail commun – en l'occurrence **un portail d'ingénierie** – peut assurer une intégration totale et une parfaite interopérabilité entre les différentes composantes du système.

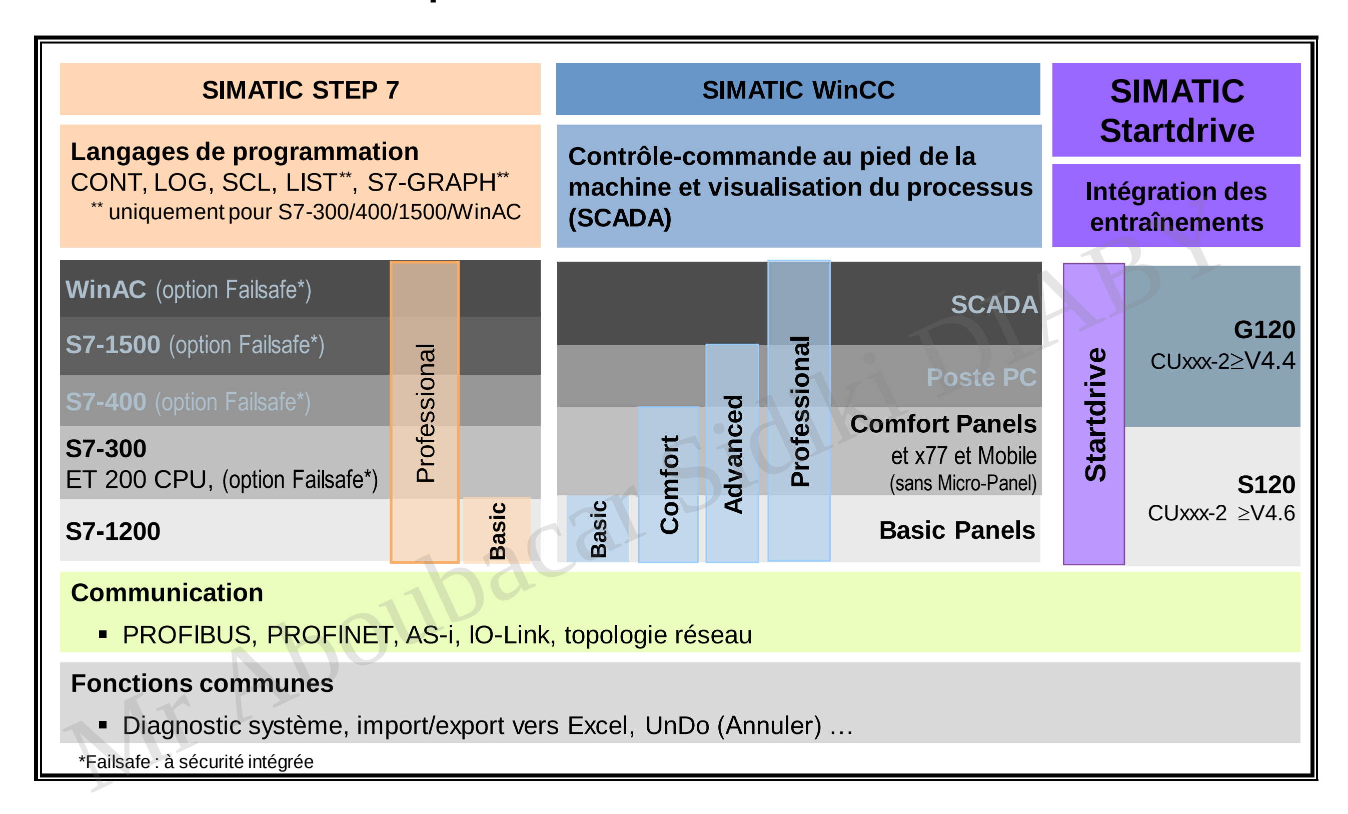
Avantages d'une plateforme d'ingénierie intégrée

- Concept de commande homogène pour toutes les tâches d'automatisation avec des services communs (par ex. configuration, communication, diagnostic)
- Cohérence automatique des données et des projets
- Bibliothèques performantes pour tous les projets d'automatisation

Les principaux produits d'ingénierie sont :

- SIMATIC STEP 7
  pour programmer les API
- SIMATIC Safety Advanced
  Sécurité sur les API
- SIMATIC WinCC
  pour configurer la visualisation des processus (contrôle-commande)
- Startdrive
  pour paramétrer les entraînements

3.4. Gamme de produits et fonctionnalités

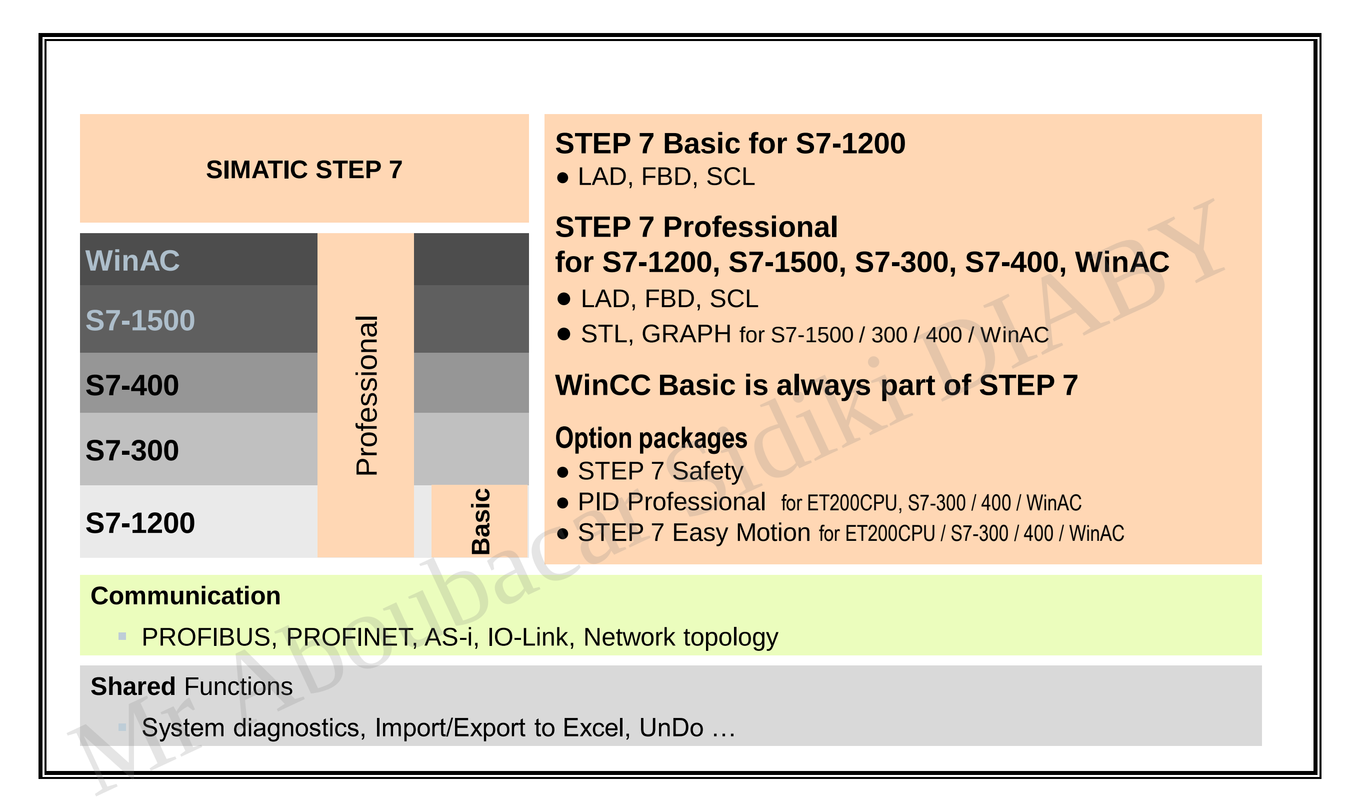


TIA Portal

Totally Integrated Automation Portal constitue un environnement de travail pour l'ingénierie intégrée avec SIMATIC STEP 7 V1x et SIMATIC WinCC V1x.

- Environnement d'ingénierie intégrée
- Cohérence automatique des données et des projets
- Ergonomie uniforme pour toutes les tâches d'automatisation
- Bibliothèques performantes pour répondre à l'ensemble des projets d'automatisation

3.4.1. Gamme de progiciels STEP 7



STEP7 Basic

Ce progiciel permet uniquement de programmer et de diagnostiquer le S7-1200.

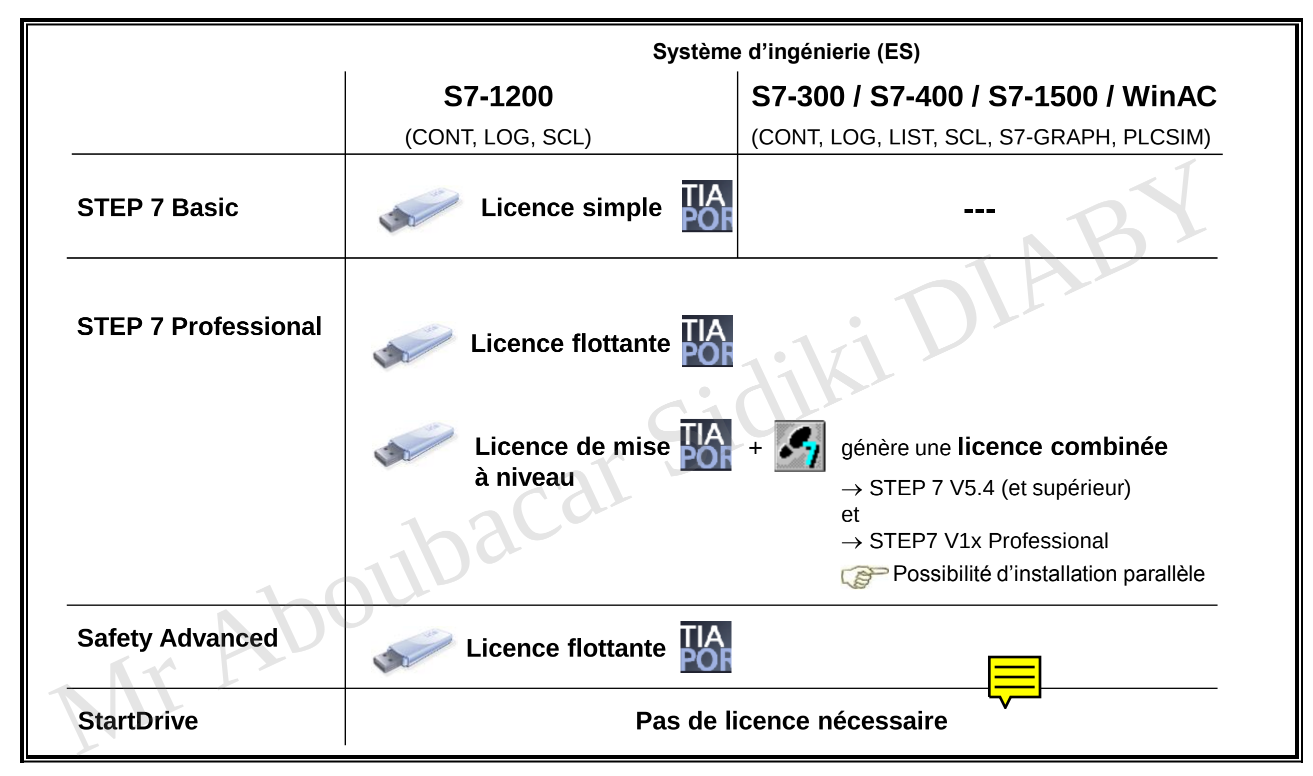
STEP7 Professional

Ce progiciel d'ingénierie permet de programmer et de diagnostiquer tous les automates de la gamme S7.

Langages de programmation

CONT, LOG et SCL disponibles pour toutes les CPU.
Avec des restrictions pour LIST et S7-GRAPH.

3.4.1.1. Licences STEP 7



	Système d'ingénierie (ES)	
	S7-1200 (CONT, LOG, SCL)	**S7-300 / S7-400 / S7-1500 / WinAC** (CONT, LOG, LIST, SCL, S7-GRAPH, PLCSIM)
STEP 7 Basic	**Licence simple**	**---**
STEP 7 Professional	**Licence flottante** **Licence de mise à niveau** + génère une **licence combinée** → STEP 7 V5.4 (et supérieur) et → STEP7 V1x Professional Possibilité d'installation parallèle	
Safety Advanced	**Licence flottante**	
StartDrive	**Pas de licence nécessaire**	

Pour programmer des automates, il suffit d'une licence (licence d'ingénierie) dans le système d'ingénierie. Le programme ne requiert pas d'autres licences sur la CPU (licence runtime).

Mr Aboubacar Sidiki DIABY

3.4.2. Gamme de progiciels WinCC

SIMATIC WinCC
pour un usage au pied de la machine
- Fonctionnalités comme WinCC flexible 2008 SP2

Fonctionnalités SIMATIC WinCC SCADA
- Fonctionnalités de base comme WinCC V7.0 SP3, diverses options RT (exécutif)
- Projets de taille moyenne : Capacité jusqu'à 64 k de Powertags (variables de process)

Fonctionnalités étendues
- F(x)-Control, OPC UA Client Serveur, pilotes tiers

SIMATIC WinCC

Contrôle-commande au pied de la machine et visualisation du processus (SCADA)

SCADA
Poste PC
Comfort Panels
et x77 et Mobile
Basic Panels
Basic
Comfort
Advanced
Professional

Communication
- PROFIBUS, PROFINET, AS-i, IO-Link, topologie réseau

Fonctions communes
- Diagnostic système, import/export vers, UnDo (Annuler) …

Ce progiciel d'ingénierie permet de configurer le contrôle-commande pour les automates qui ont été programmés.

WinCC Basic et Comfort

Configuration de pupitres opérateur au pied de la machine

WinCC Advanced

Configuration de pupitres opérateur ou d'un terminal d'exploitation sur base PC au pied de la machine

WinCC Professional

SCADA (**S**upervisory **C**ontrol **a**nd **D**ata **A**cquisition)

Il s'agit d'assurer la supervision et la visualisation de processus techniques à l'aide d'un système informatique.

3.4.2.1. Licences WinCC

	Système d'ingénierie (ES)	Runtime (RT)
WinCC Basic	--- (composante de STEP7) **Licence flottante**	--- (composante des pupitres)
WinCC Comfort	**Licence flottante** **Licence de mise à niveau** + génère une **licence combinée** → WinCC flexible 2008 Standard et → WinCC Comfort Possibilité d'installation parallèle	--- (composante des pupitres)
WinCC Advanced	**Licence flottante** **Licence de mise à niveau** + génère une **licence combinée** → WinCC flexible 2008 Advanced et → WinCC Advanced Possibilité d'installation parallèle	**Licence simple** **Licence de mise à niveau** →
WinCC Professional	**Licence flottante** Licence de mise à niveau non disponible car pour WinCC V7 (et supérieur), il existe un autre modèle de licence Installation parallèle WinCC V7 avec WinCC Professional actuellement encore impossible	**Licence simple** Licence de mise à niveau non disponible car pour WinCC V7 (et supérieur), il existe un autre modèle de licence

Généralement, WinCC Basic fait partie du système d'ingénierie STEP7. Il existe toutefois également une licence séparée pour WinCC Basic au cas où le PC de configuration ne serait pas équipé de STEP7.

Pour la configuration de la visualisation, il faut une licence appropriée dans le système d'ingénierie (licence d'ingénierie).

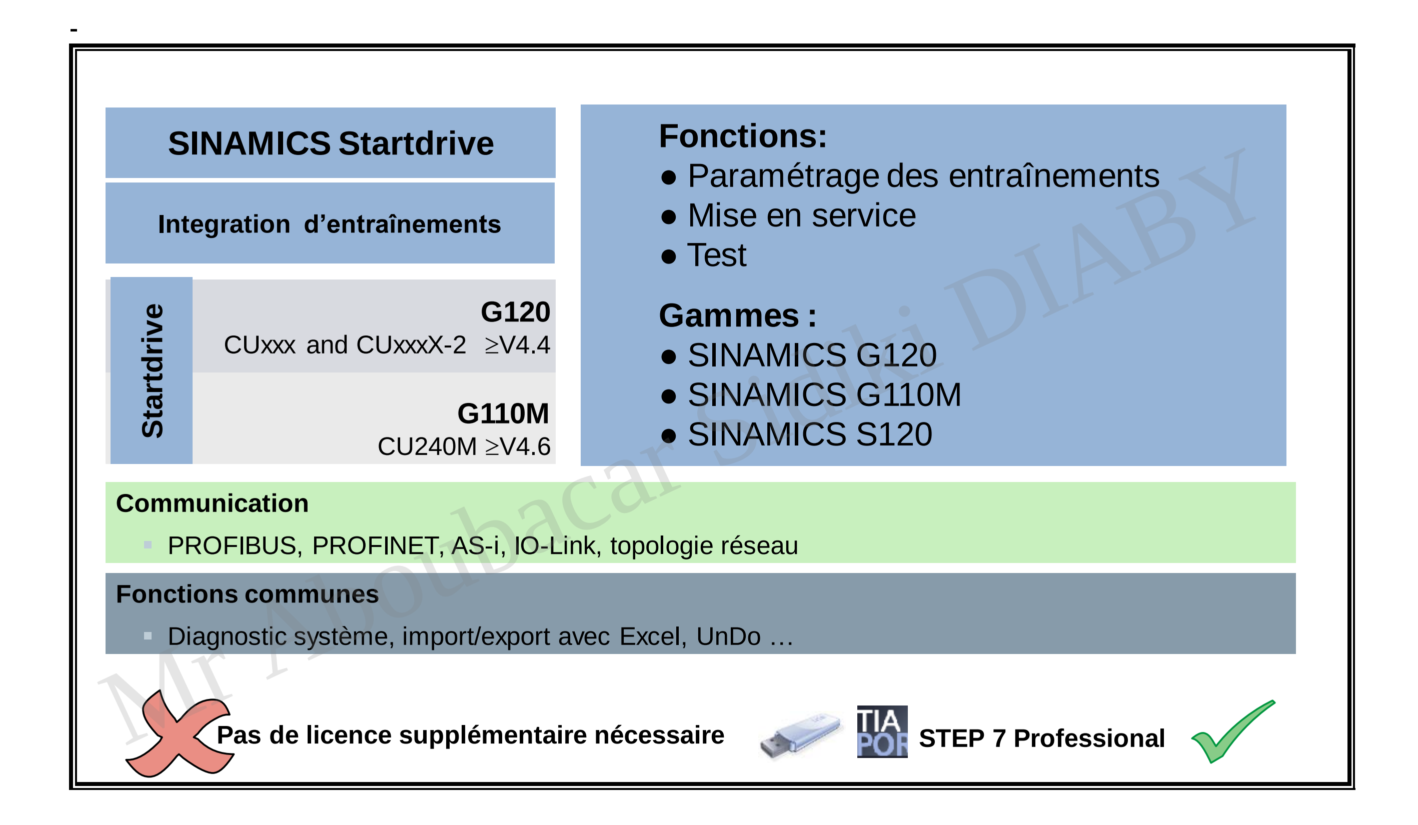
Pour les terminaux d'exploitation sur base PC, la configuration en cours sur le terminal requiert également une licence d'exécution (licence runtime).

Modèle de licence WinCC V7 (et versions supérieures)

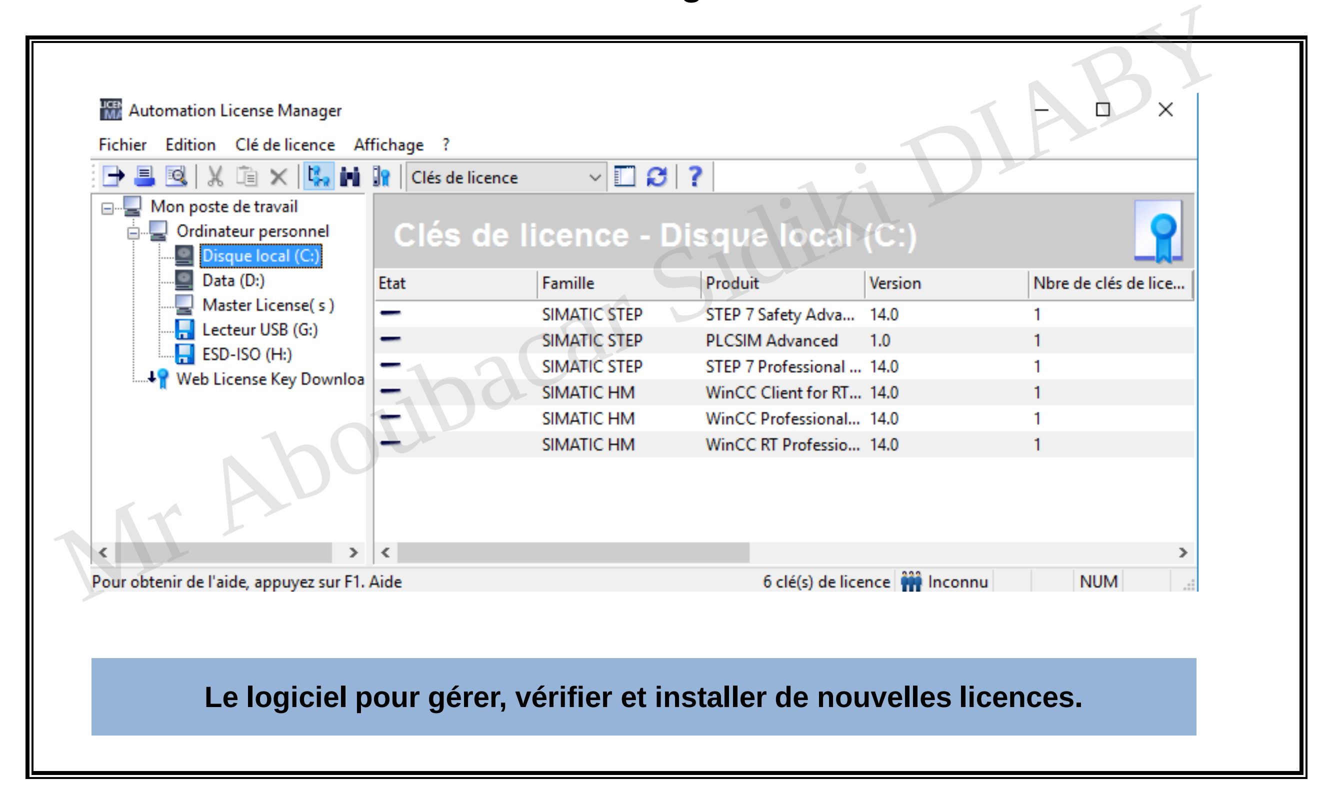
Il existe une licence de configuration (licence RC), qui couvre l'exécution (licence RT) et l'ingénierie (licence ES). Il n'existe pas à ce jour de licence de mise à niveau directe, mais il y a des packs logiciels qui permettent de passer au niveau supérieur.

3.4.3. Gamme de produits Startdrive et licences

-

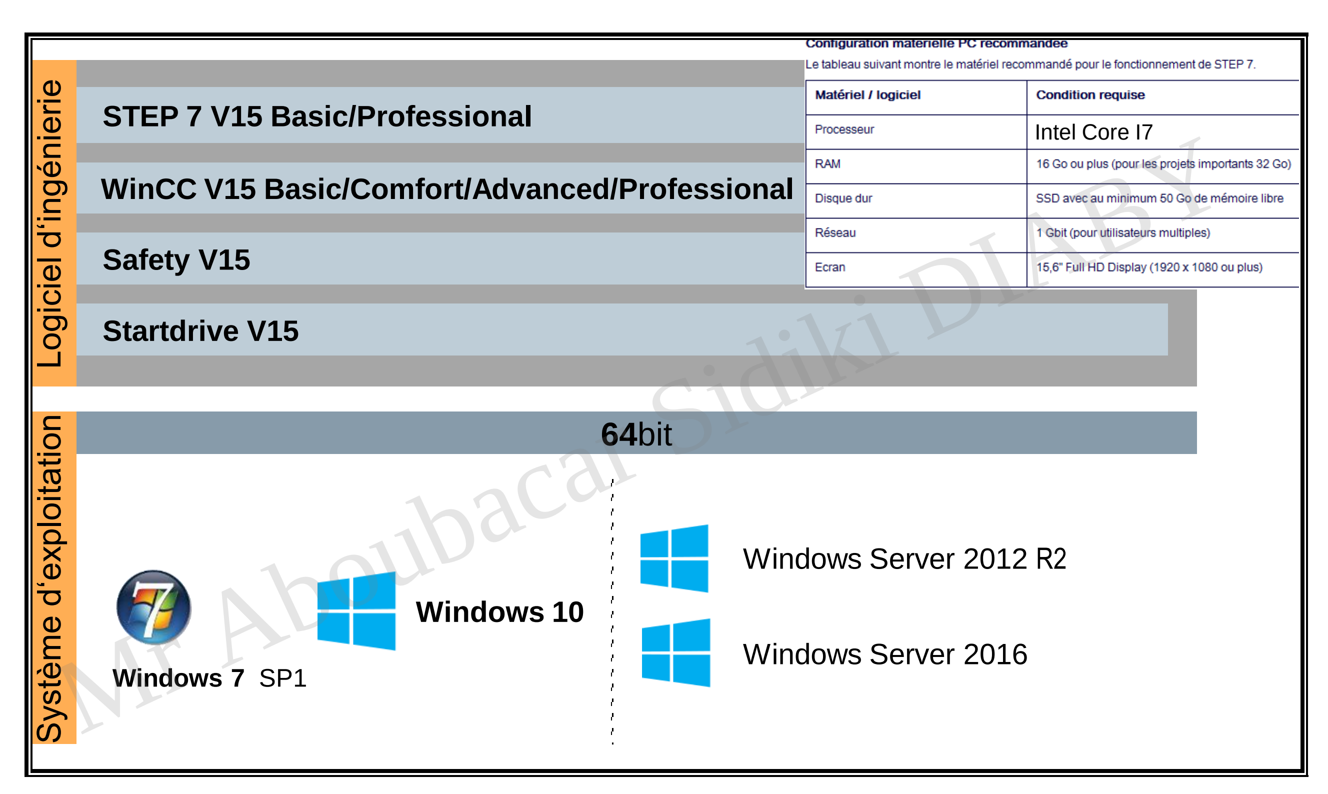


3.5. Automation License Manager



Automation License Manager fait partie de l'installation logicielle SIMATIC et gère la gestion des licences du logiciel SIMATIC.

3.5.1. Systèmes d’exploitation pour PC/PG

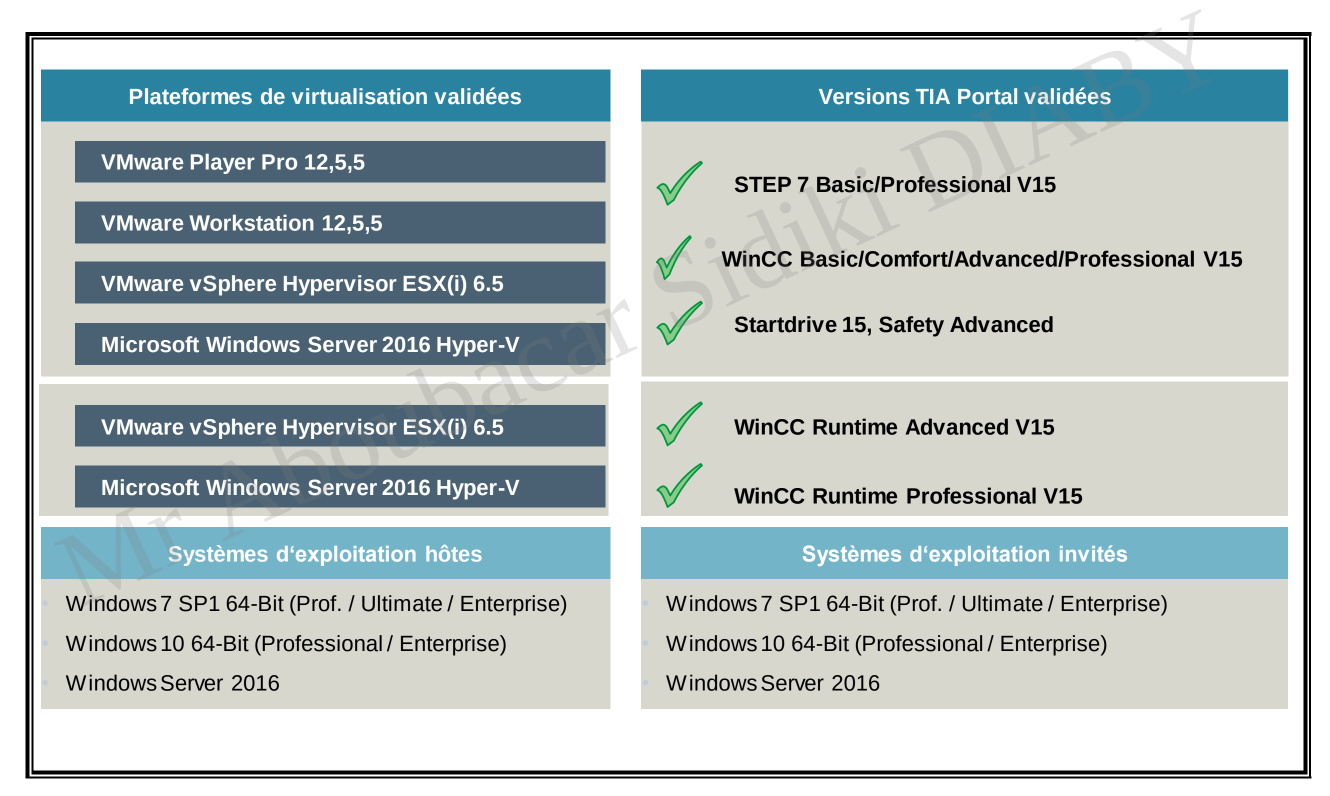


Configuration matérielle PC recommandée

Le tableau suivant montre le matériel recommandé pour le fonctionnement de STEP 7.

Matériel / logiciel	Condition requise
Processeur	Intel Core I7
RAM	16 Go ou plus (pour les projets importants 32 Go)
Disque dur	SSD avec au minimum 50 Go de mémoire libre
Réseau	1 Gbit (pour utilisateurs multiples)
Ecran	15,6" Full HD Display (1920 x 1080 ou plus)

3.5.2. Virtualisation (logiciels validés)



3.5.3. Installation « côte à côte »



Plusieurs TIA-Portal peuvent être installés sur le même PG/PC

A partir de la v14, il est nécessaire d'avoir un système 64 bits

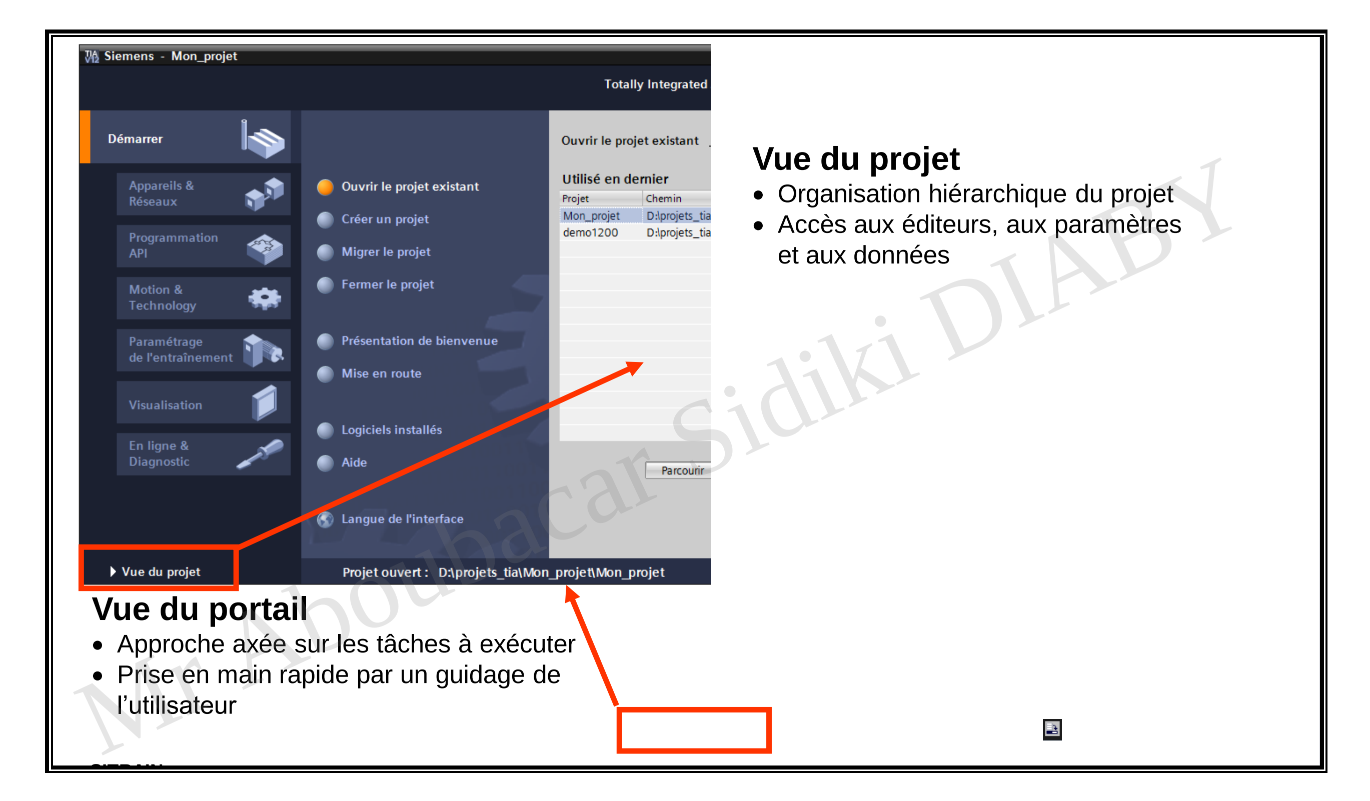
Compatibility Tool
est disponible comme outil pour vérifier les installations logicielles parallèles,

Sous Aide du TIA Portal lancer:
TIA Portal Information Center > Tools & Apps > Configurators
ou
Avec l'article ID: 64847781
du Support

Compatibility Tool pour Automation et Drives

Avec Compatibility Tool, vous pouvez vérifier la compatibilité des différentes versions logicielles SIMATIC, soit via le TIA Portal Information Center ou sur le Support (https://support.industry.siemens.com) sous ID: 64847781.

3.6. Vues du système d'ingénierie



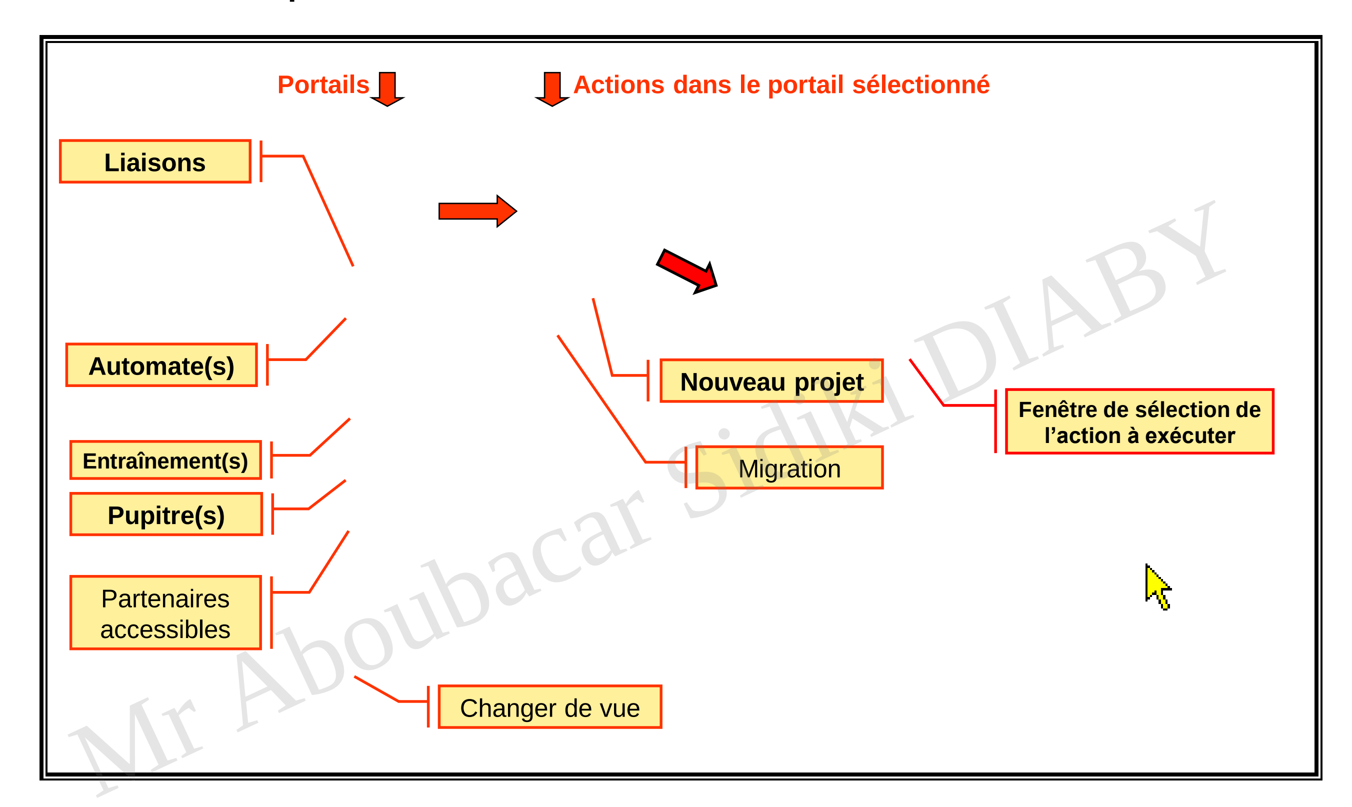
Vue du portail

- Approche axée sur les tâches à exécuter
- Prise en main rapide par un guidage de l'utilisateur

Vue du projet

- Partage hiérarchique du projet
- Les éditeurs requis s'ouvrent en fonction des tâches à réaliser
- Les éditeurs, les paramètres et les données sont visualisés dans une seule et même vue

3.6.1. Vue du portail



Présentation de la vue du portail :

- Chaque portail permet de traiter une catégorie de tâches
- La fenêtre affiche la liste des actions pertinentes pour la tâche sélectionnée
- Une fenêtre de sélection propose l'ensemble des actions à exécuter

Portails

Accès aux appareils, aux composantes et à leurs liaisons.

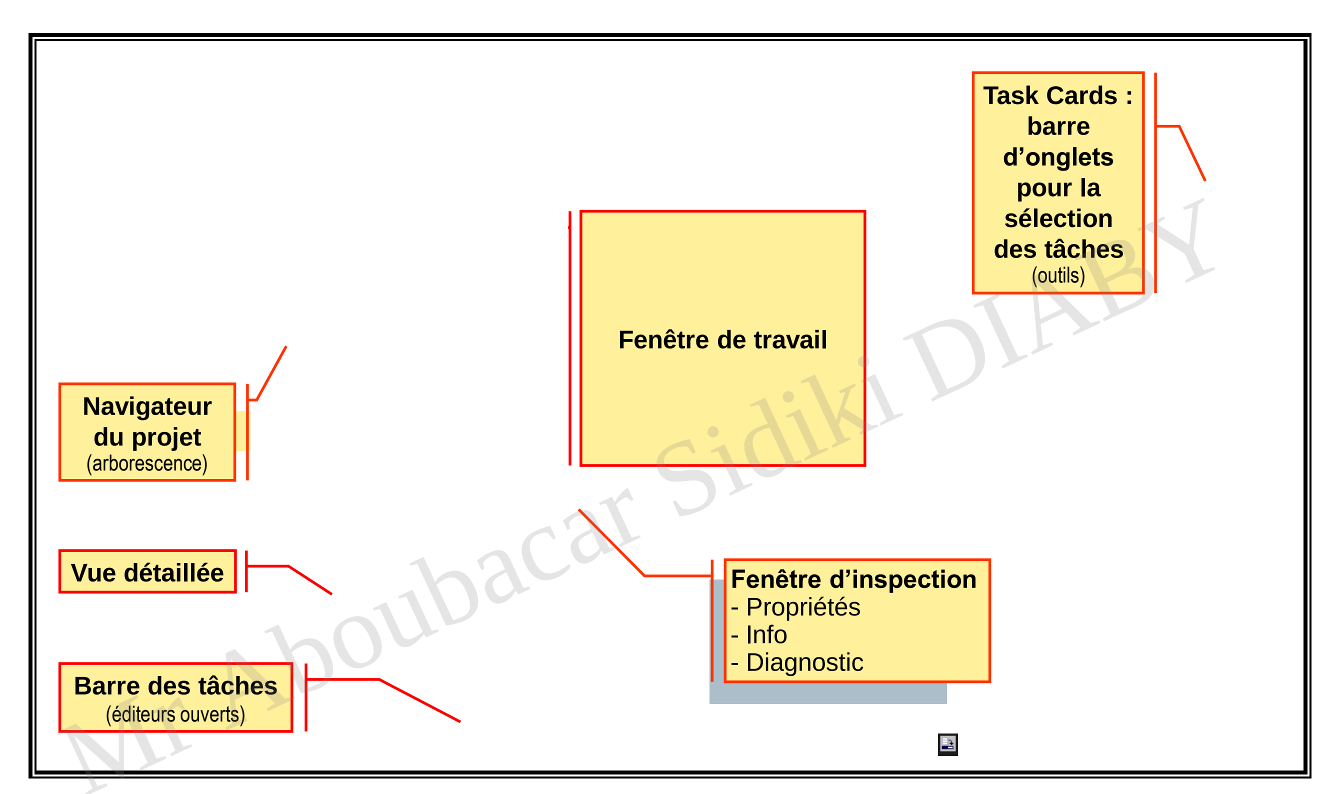
Actions

Les actions pouvant être exécutées dans le portail sélectionné sont affichées. Le portail est doté d'une aide contextuelle.

Fenêtre de sélection

La fenêtre de sélection est disponible sur chacun des portails. Le contenu de la fenêtre est fonction de la sélection courante.

3.6.2. Vue du projet



Navigateur du projet

Le navigateur du projet contient toutes les composantes et les données du projet permettant de développer une solution d'automatisation. Toutes les composantes et tous les éditeurs disponibles pour le projet sont affichés et peuvent être ouverts dans l'arborescence du navigateur. A chaque éditeur correspond une icône qui permet d'identifier les objets qui y sont associés.

Le navigateur du projet n'affiche que les éléments qui peuvent être gérés par l'appareil sélectionné.

Fenêtre de travail

La fenêtre de travail permet de visualiser les objets qui ont été ouverts pour être traités. Il peut s'agir de composants matériels, de blocs, de tables de variables pour des API, de vues pour des pupitres opérateurs (HMI), etc.

Si plusieurs objets sont ouverts simultanément, ils sont affichés dans la barre des tâches sous forme d'onglets.

Task Cards (barre d'onglets)

Le contenu des menus affichés dans la barre d'onglets pour la sélection des tâches (task cards) est fonction de l'objet sélectionné dans la fenêtre de travail.

S'il s'agit par exemple d'une station matérielle, le catalogue du matériel est disponible en onglet.

S'il s'agit d'un bloc de programme, l'utilisateur pourra visualiser les instructions.

Si une vue est ouverte, les objets de configuration, les modèles d'animation et les outils de mise en page sont proposés.

Fenêtre d'inspection

La fenêtre d'inspection permet de visualiser des informations complémentaires sur un objet sélectionné ou sur les actions en cours d'exécution.

Elle permet également de traiter les propriétés disponibles des objets sélectionnés (par ex. des vues, des objets de vue, des variables).

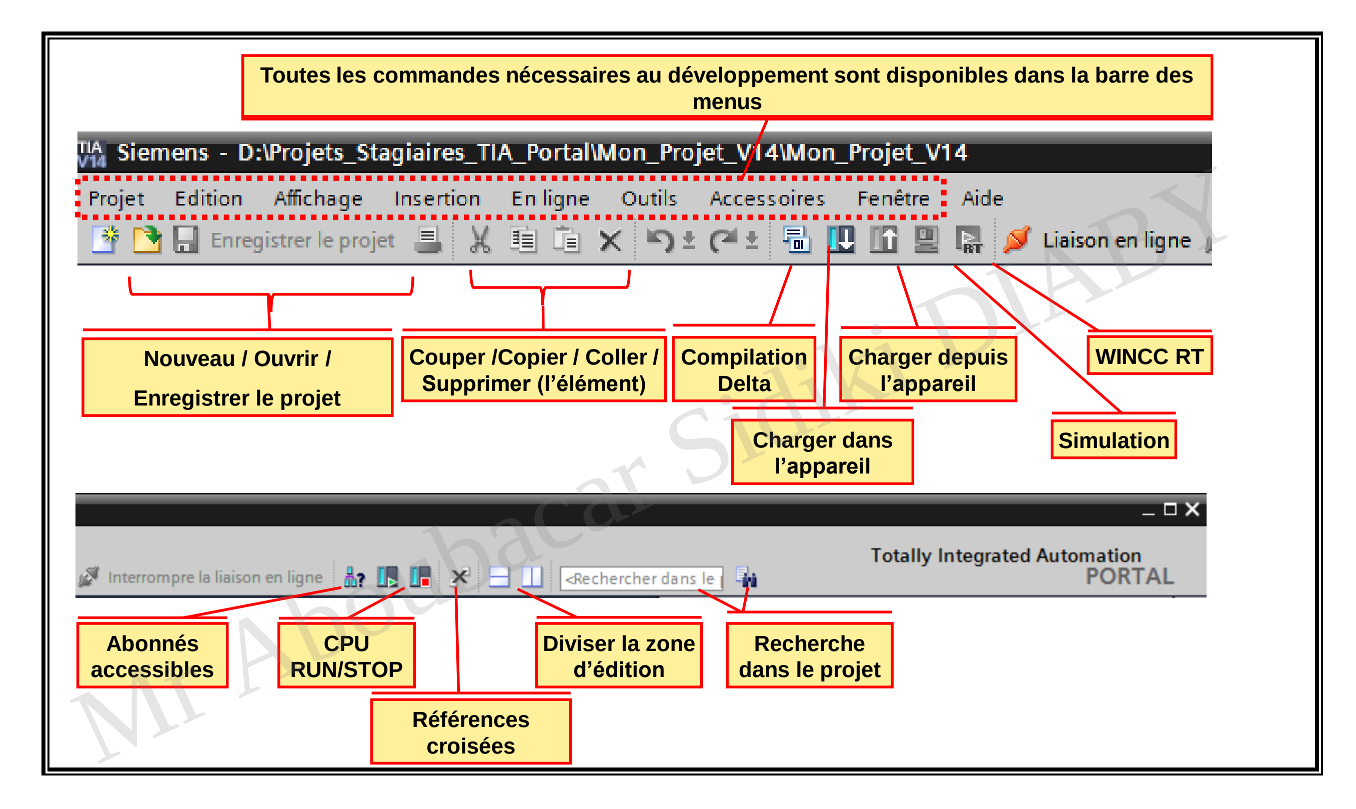
Tous les messages du système d'ingénierie s'affichent dans la fenêtre d'inspection, par ex. les messages générés lors de la création d'un projet.

☞ Il convient de vérifier systématiquement le contenu de cette fenêtre après avoir créé un nouveau projet pour contrôler la présence éventuelle de messages d'erreur ou d'avertissement.
Menu : > *Affichage* > *Fenêtre d'inspection*

Vue détaillée

La vue détaillée est une fenêtre auxiliaire. Elle contient des éléments de l'objet de configuration sélectionné dans le navigateur du projet. Ces éléments peuvent être utilisés dans la fenêtre de travail active (en les déplaçant dans la fenêtre de travail par glisser-déposer), ce qui permet d'accéder plus rapidement aux objets requis (par ex. les variables).

3.6.3. Barre des menus et barre des outils



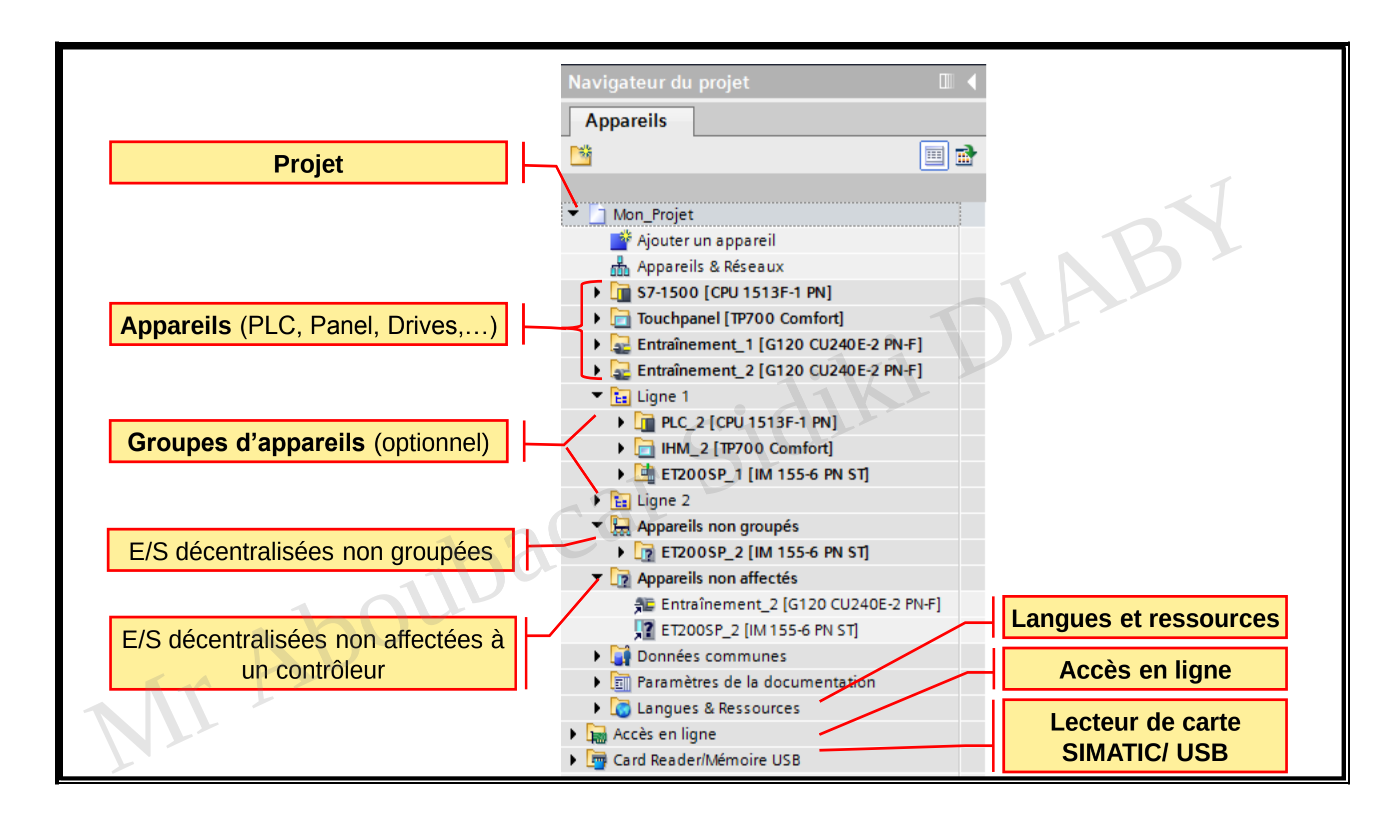
Barre des menus

La barre des menus contient l'ensemble des commandes nécessaires à votre développement.

Barre des outils

La barre des outils contient les boutons des commandes fréquemment utilisées. Ainsi elles sont accessibles rapidement.

3.6.4. Navigateur de projet (premier niveau)



La fenêtre « Navigateur du projet » donne accès à l'ensemble des composantes et des données du projet. Toutes les composantes et tous les objets disponibles sont affichés dans une arborescence et peuvent être ouverts par un double clic.

Les actions suivantes peuvent être exécutées :

- Ajout de nouvelles composantes (automates, pupitres opérateur, etc.)
- Traitement des composantes existantes
- Interrogation et modification des propriétés des composantes existantes
- Diagnostic des composantes accessibles

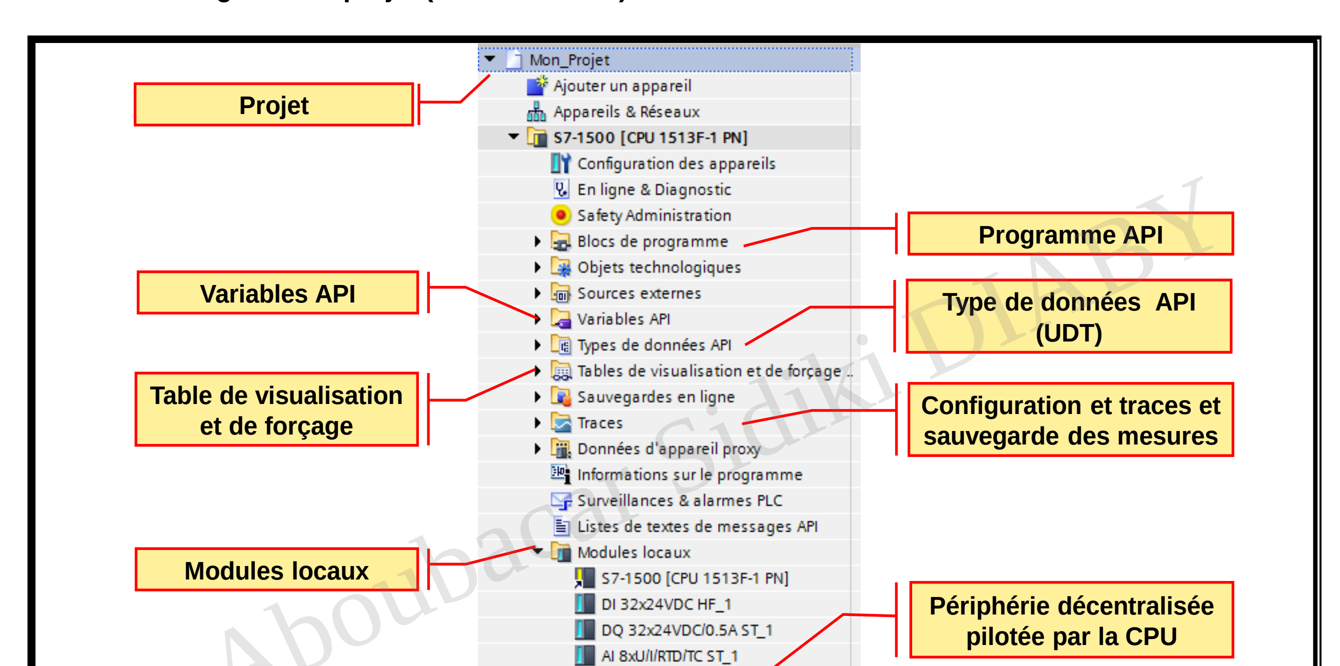
Pour améliorer la lisibilité, les objets peuvent être regroupés (des stations complètes ou l'ensemble des blocs d'une station).

Les E/S décentralisées venant d'être insérées sont mémorisées dans le dossier « Appareils non groupés » et peuvent être déplacées vers les groupes que vous avez créés.

Un lien vers un module d'E/S décentralisées se trouve dans le dossier « Appareils non affectés » jusqu'à son assignation à un contrôleur ou un maître.

Les dossiers « Données communes », « Paramètres de la documentation » et « Langues & Ressources » font référence au projet; les dossiers « Accès en ligne » et « Card Reader/Mémoire USB » sont indépendants du projet.

3.6.4.1. Navigateur de projet (second niveau)



Pour une meilleure vue d'ensemble, les blocs peuvent être affectés à des groupes que vous avez créés. Ce groupement sert uniquement à la lisibilité du programme et n'a pas d'impact sur l'exécution du programme. Cette information n'est pas chargée dans la CPU.

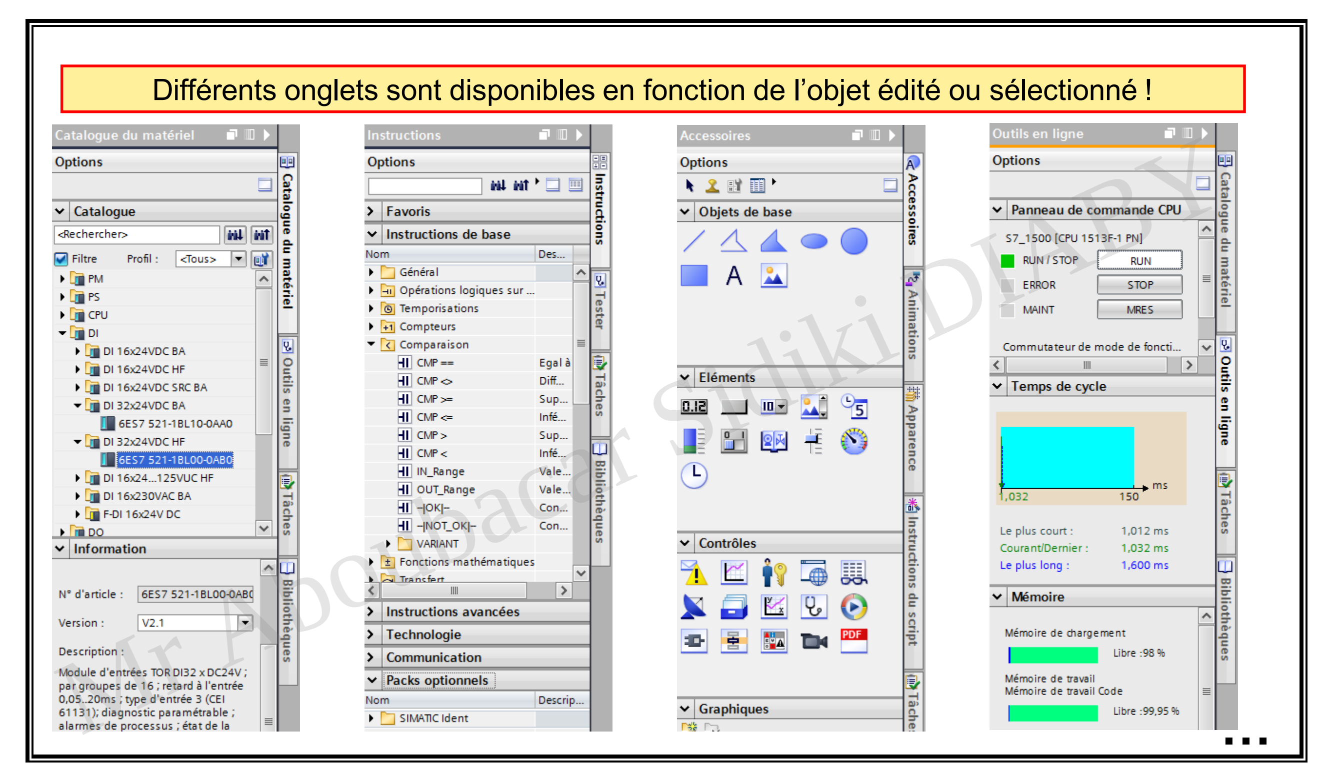
Le dossier « Modules locaux » contient l'ensemble des modules centralisés.

Un device ou un esclave affecté à un contrôleur ou à un maître se retrouve dans le dossier « Périphérie décentralisée » du contrôleur ou du maître concerné.

Afficher ou masquer une branche de l'arborescence

Pour afficher ou masquer une branche de l'arborescence, il suffit de cliquer sur le triangle ▸. En cliquant sur le triangle, l'arborescence sous-jacente peut être affichée ▸→▾ ou masquée ▾→▸.

3.6.5. Task Cards (Barres d'onglets)

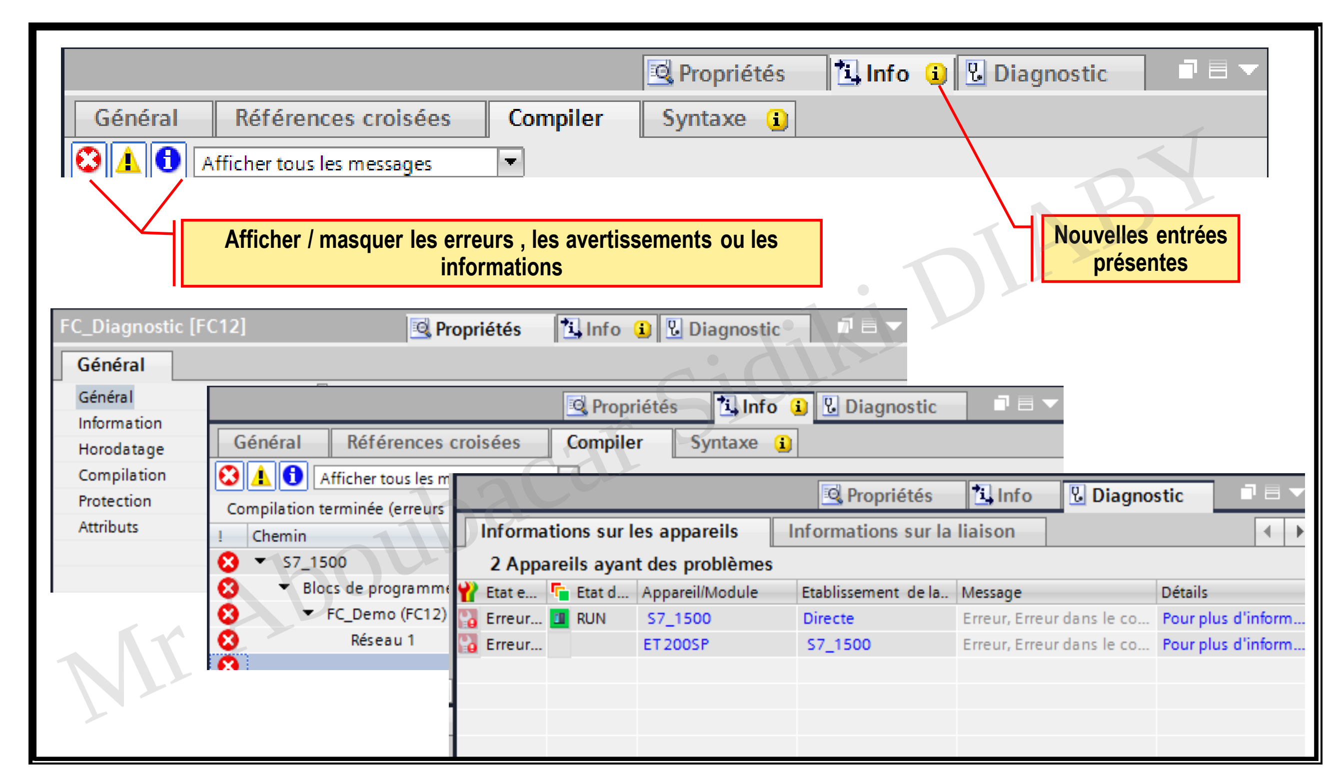


Les onglets (Task Cards) affichés sur la droite de l'écran peuvent être ouverts ou fermés à tout moment.

Les onglets disponibles à l'écran sont fonction des produits installés et de l'objet ouvert dans la fenêtre de travail ou en cours de traitement. S'ils ne sont pas tous visibles, utilisez les boutons fléchés situés en bas à droite pour faire défiler la barre d'onglets. Ces onglets permettent d'exécuter différentes actions :

- **Catalogue du matériel**
  Il contient les composantes matérielles disponibles, comme la CPU, les modules, etc.
- **Instructions**
  Instructions pour la programmation de blocs ;
  modèles de programme et assistant avec liste de fonctions pour la programmation avec des scripts (VBS ou scripts C pour WinCC Professional)
- **Outils en ligne**
  Lorsqu'une liaison en ligne est établie, des informations en ligne et des données de diagnostic peuvent être consultées (comme le temps de cycle actuel de la CPU ou la configuration de la mémoire de travail et de chargement dans la CPU). Par ailleurs, la CPU peut être basculée en mode STOP et RUN.
- **Accessoires**
  Objets configurables (graphiques, éléments d'affichage et de commande) dans différentes palettes (objets de base, éléments, contrôles, contrôles spécifiques à l'utilisateur, graphiques)
- **Animations**
  Modèles pour la dynamisation d'objets graphiques dans différentes palettes (mouvements, affichage, dynamisation de variables)
  ☞ Dans WinCC Basic, il n'y a pas d'onglet « Animations ».
- **Apparence**
  Les outils d'adaptation de la disposition des vues lors de la configuration des pupitres opérateur (zoom, hiérarchisation des niveaux, disposition de la trame, objets hors zone)

3.6.6. Fenêtre d'inspection



La fenêtre d'inspection permet de visualiser des informations supplémentaires sur un objet sélectionné ou sur des actions en cours d'exécution. La fenêtre d'inspection comprend les onglets suivants :
→ sélection en cliquant sur les onglets


Ce symbole dans l'onglet indique la présence de nouvelles entrées.

En cas d'affichage d'erreurs, vous pouvez sauter vers l'emplacement de l'erreur ou dans l'éditeur correspondant en double-cliquant sur l'information d'erreur.

Menu « Propriétés »

Cette option de menu permet de visualiser les propriétés de l'objet sélectionné et de modifier les propriétés éditables.

Menu « Info »

Cet onglet permet d'afficher des informations complémentaires sur l'objet sélectionné. Par ailleurs, des messages sur les actions en cours d'exécution sont affichés (par ex. sur la compilation ou le chargement de blocs dans la CPU).

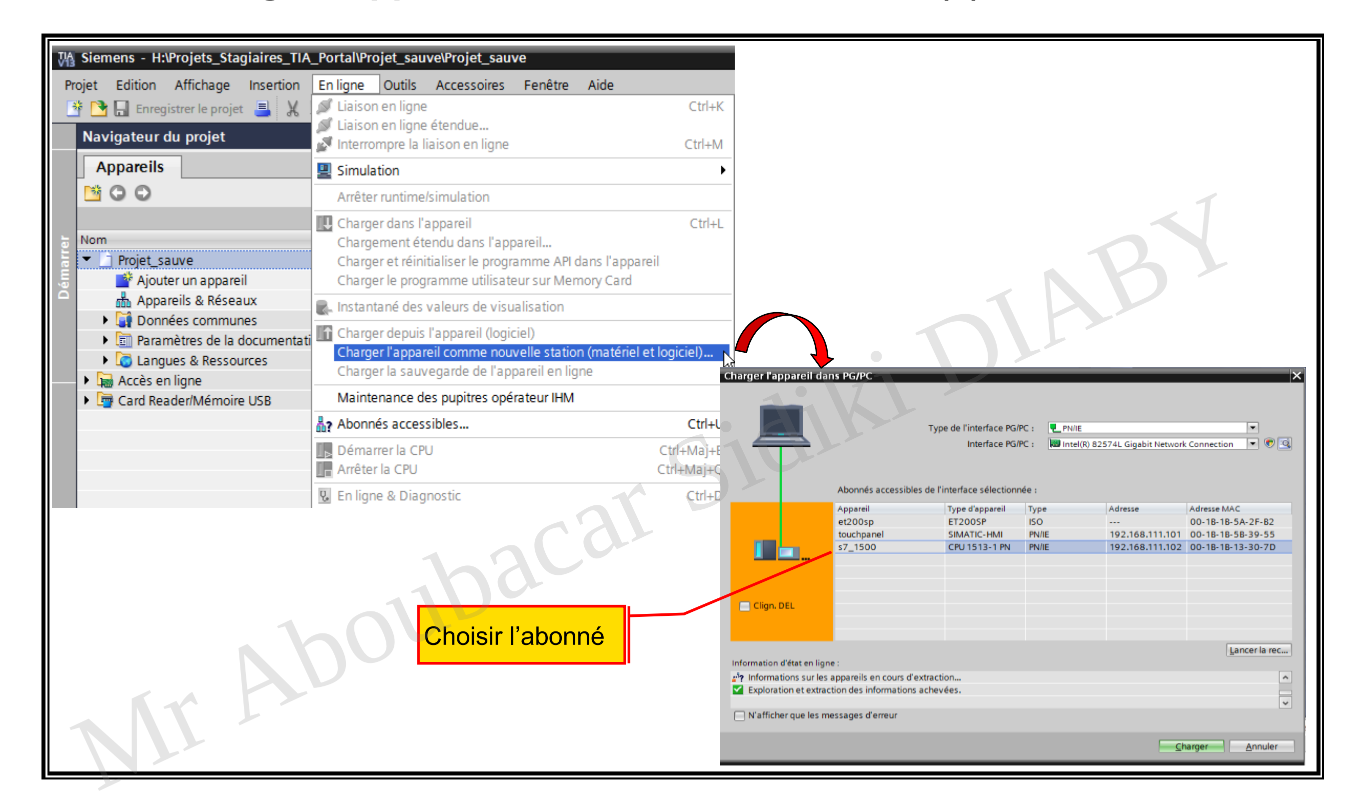
Onglet « Général » → Informations générales
Onglet « Références croisées » → Affichage des occurrences actuelles de l'objet sélectionné
Onglet « Compiler » → Etat du processus de compilation
Onglet « Syntaxe » → Informations sur les instructions de programmation non valides

Menu « Diagnostic »

Cet onglet permet d'afficher des informations sur le diagnostic système et les événements d'alarme configurés.

Onglet « Affichage des alarmes » → Affichage des alarmes en instance sur la CPU (ex. Alarm_S)
Onglet « Informations sur les appareils »
Onglet « Informations sur la liaison » → pour les composantes qui sont actuellement en ligne

3.7. Charger l’appareil comme nouvelle station (1)

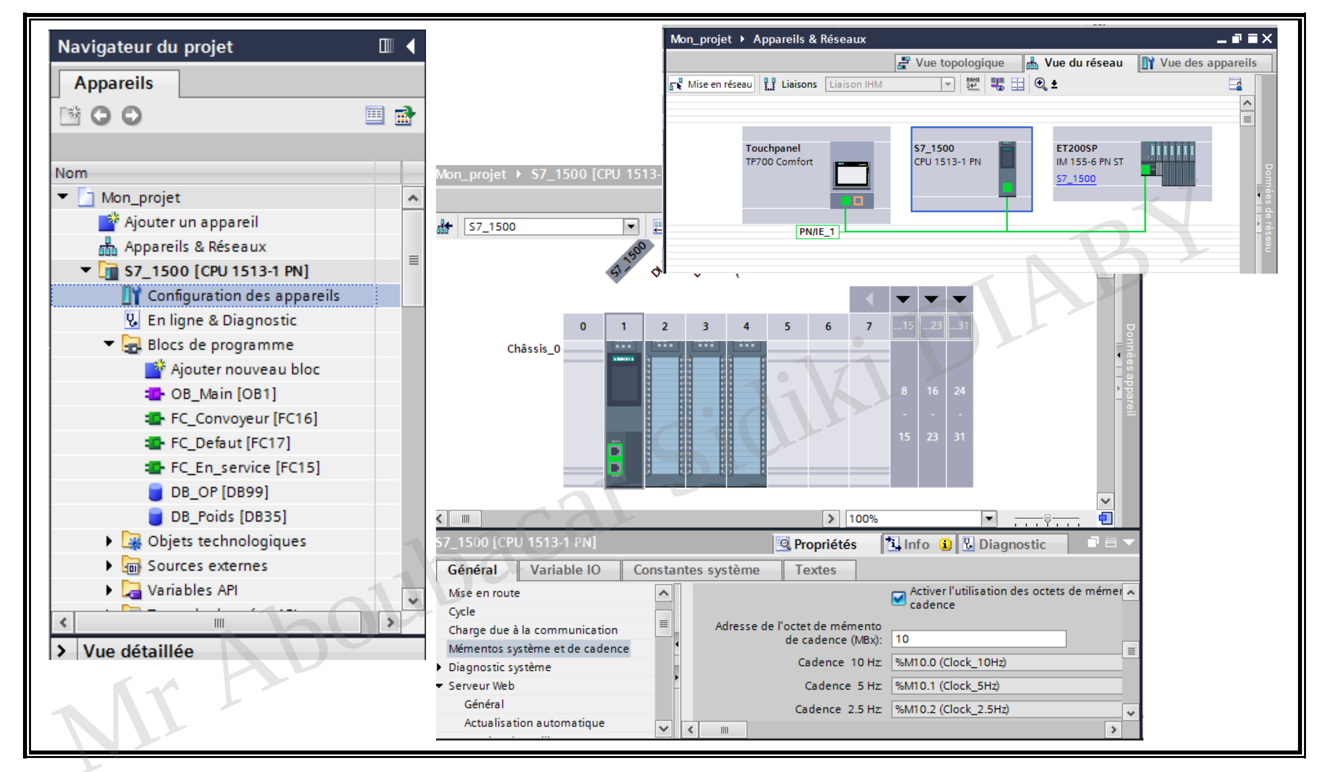


La fonction « charger l’appareil comme nouvelle station (matériel et logiciel) » permet d’importer la station API avec son paramétrage et ses commentaires.
Cette fonction est indispensable si la station en mode hors ligne ne se trouve pas sur la PG.
Cette station importée dans un nouveau projet peut être modifiée, sauvegardée et chargée dans la CPU

Pré-requis

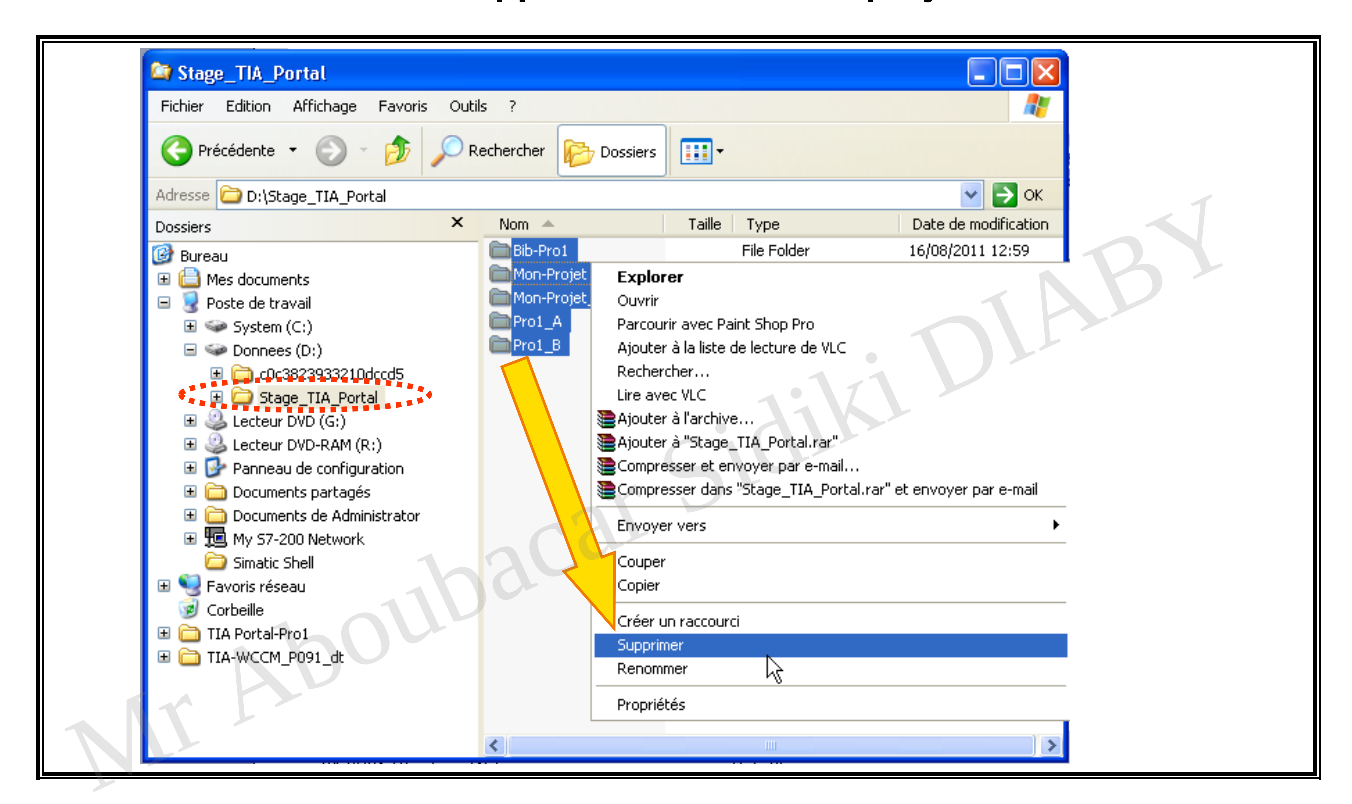
La station doit être paramétrée.

3.7.1. Charger l'appareil comme nouvelle station (2)



Après cette procédure c'est l'ensemble de la station (matériel centralisé et décentralisé avec paramétrage ainsi que programme complet avec commentaires et symboliques) qui est mis à disposition de l'utilisateur pour traitement ultérieur.

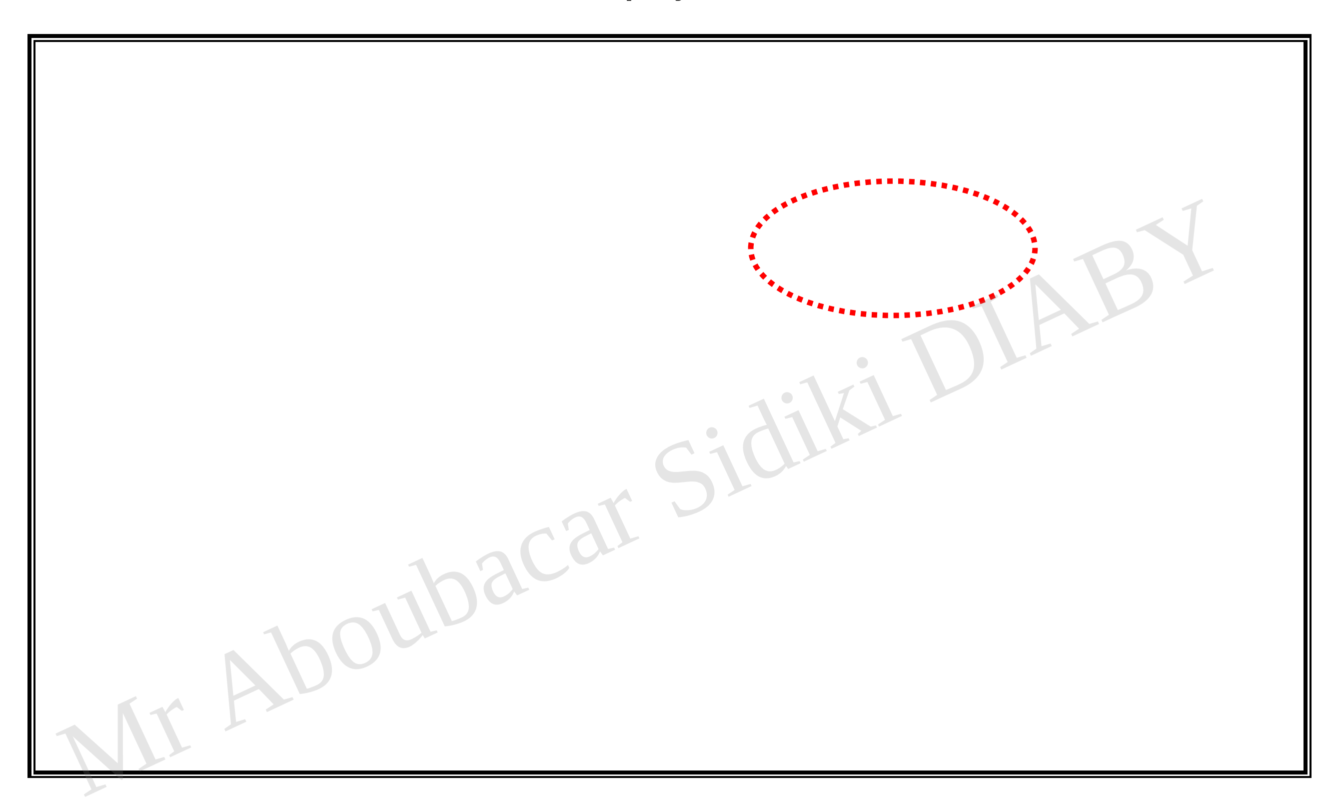
3.8. Exercice 1 : Suppression d’anciens projets



Enoncé et marche à suivre

Dans le dossier [Lecteur]:\Stage_TIA_Portal, supprimez tous les fichiers et dossiers (voir figure).

3.8.1. Exercice 2 : Créer un nouveau projet



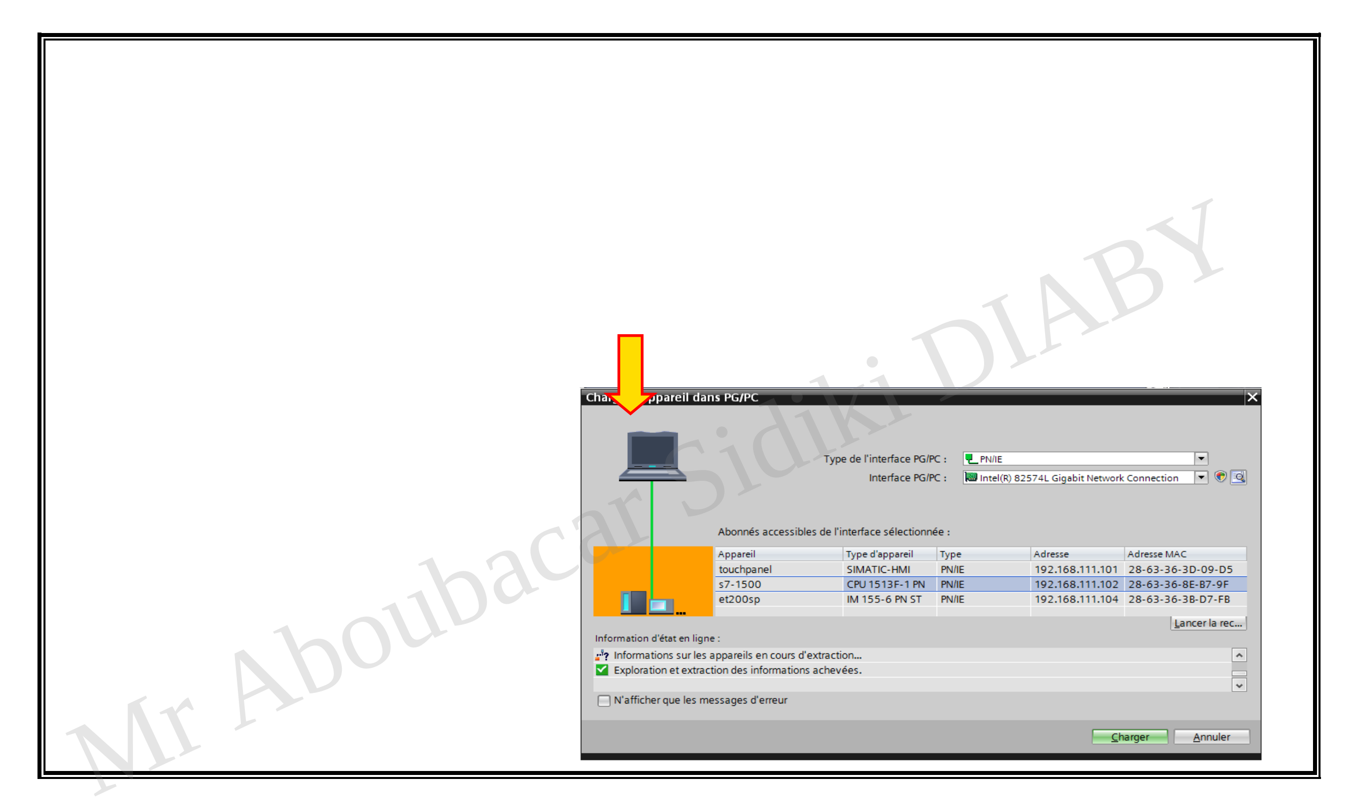
Enoncé

Créez un nouveau projet portant le nom « Mon_projet ».

Marche à suivre

1. Activez « Créer un projet » dans la vue du portail.
2. Entrez le nom du projet et le chemin comme indiqué dans la figure ci-dessus et créez le projet en cliquant sur « Créer ».

3.8.2. Exercice 3: Chargement de la configuration actuelle



Enoncé

Dans votre projet, créez une nouvelle station en chargeant la configuration et les paramètres actuels du S7-1500 dans votre projet.

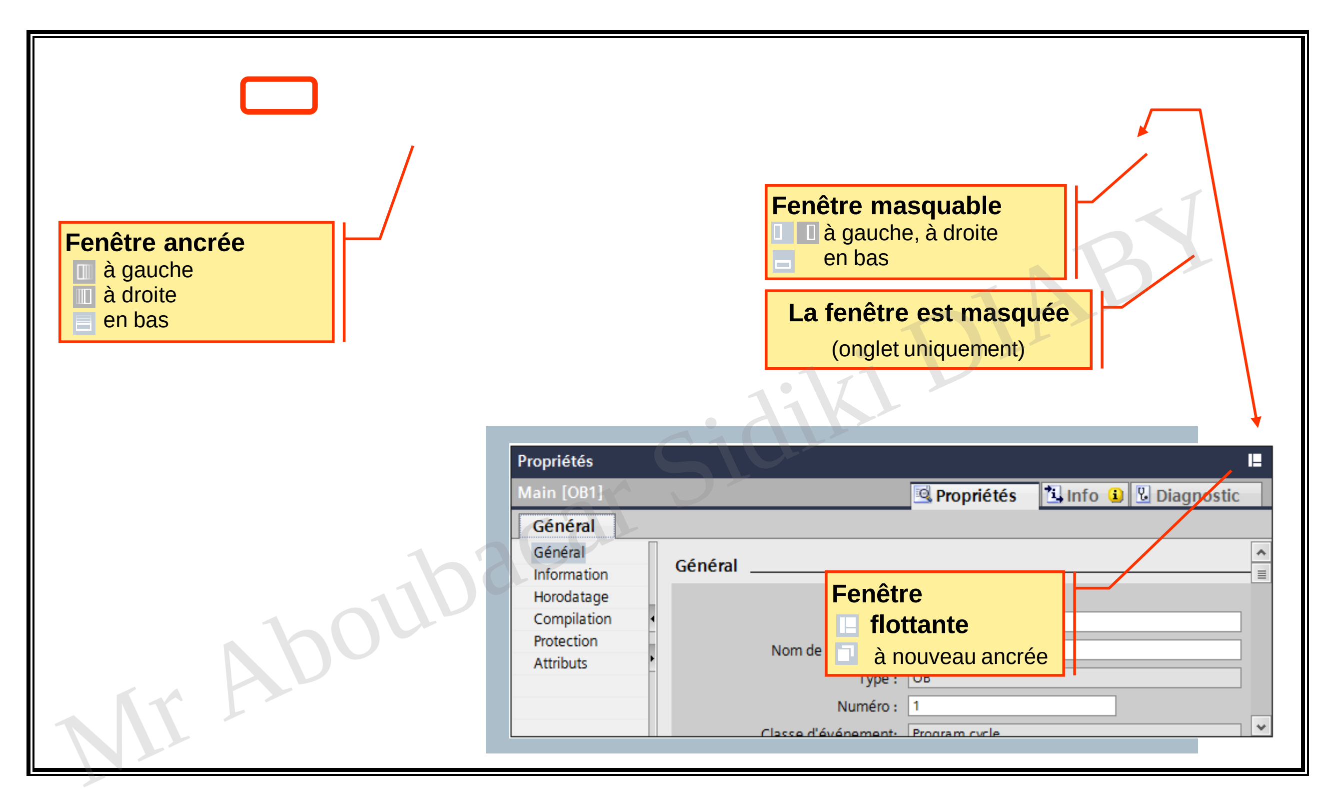
Marche à suivre

1. Basculez dans la « Vue du projet ».
2. Sélectionnez votre projet et sous « En ligne » activez la fonction « Charger l'appareil comme nouvelle station (matériel et logiciel)… ».
3. Dans le dialogue « Charger l'appareil dans PG/PC », sélectionnez le S7-1500 et lancez le chargement par « Charger ».
4. Sauvegarder votre projet.

Résultat

La vue des appareils montre la configuration récupérée.

3.9. Disposition des fenêtres

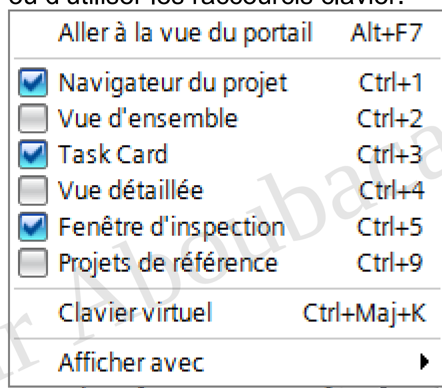


La position et le comportement des fenêtres à l'écran peuvent être configurés individuellement. Les fenêtres rarement utilisées peuvent être masquées pour optimiser la place disponible dans la fenêtre de travail.

La configuration actuelle de l'interface d'ingénierie est enregistrée dans le profil utilisateur Windows. Lorsque le projet est enregistré, la position et le comportement des fenêtres sont enregistrés automatiquement.

Menu → Affichage

Pour ouvrir, fermer ou réduire une fenêtre, il suffit d'activer les fonctions du menu « Affichage » ou d'utiliser les raccourcis clavier.



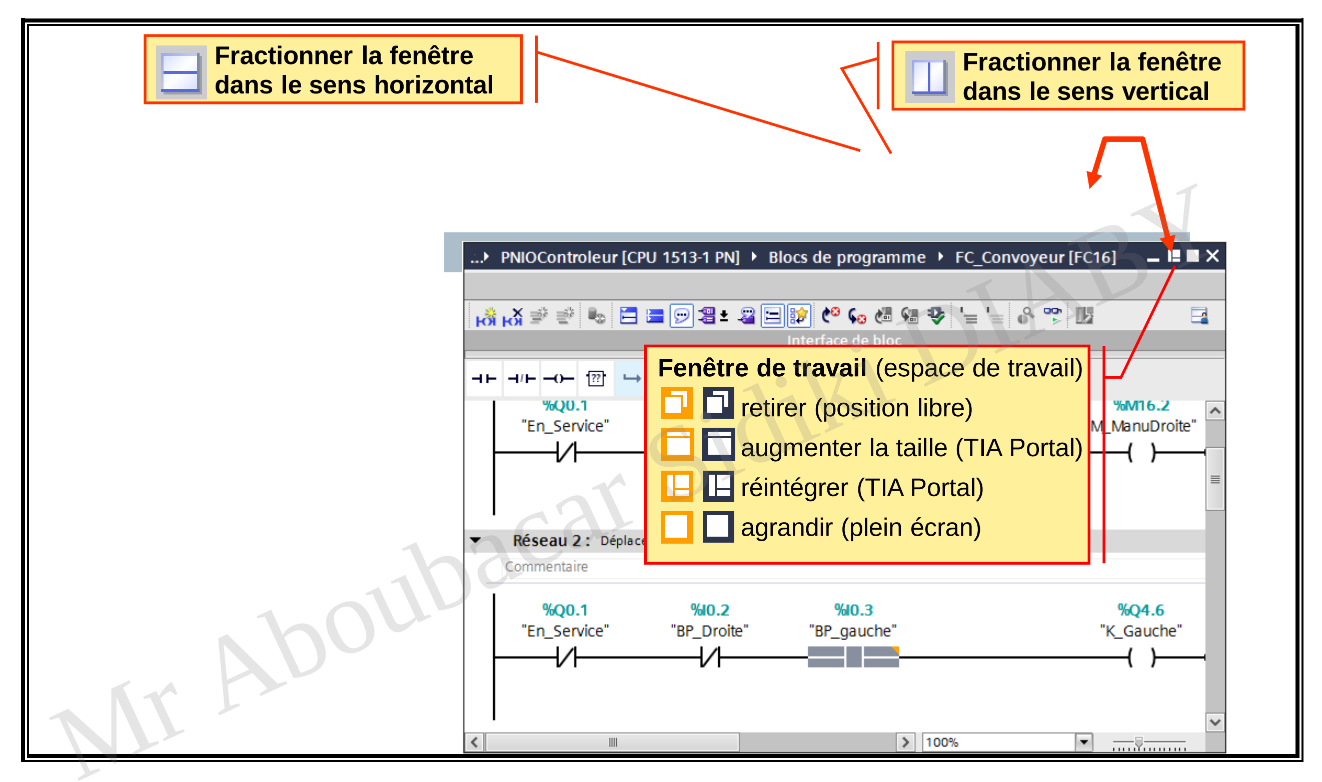
Configuration des fenêtres

- Fenêtre ancrée
  - La position et la taille de la fenêtre dans l'interface sont fixes.
  - Possibilité de positionner la fenêtre à gauche, en bas ou à droite en dehors de la fenêtre de travail
  - La fenêtre est toujours ouverte, elle réduit l'espace de la fenêtre de travail.
- Fenêtre masquable
  - Fenêtre masquée sur le bord de l'interface
  - Possibilité de positionner la fenêtre à gauche, en bas ou à droite, fondu à l'ouverture sur l'espace de travail
  - Etat standard = Fenêtre fermée et onglet affiché sur le bord de l'interface
  - Ouverture de la fenêtre en cliquant sur l'onglet
  - Fermeture automatique lors du clic suivant en dehors de la fenêtre
- Fenêtre flottante **!! Intéressant pour un 2e moniteur !!**
  - Possibilité de positionner librement la fenêtre à l'intérieur de l'interface
  - La fenêtre recouvre en permanence la zone d'affichage située en dessous.

En cliquant sur les fonctions dans la barre de titre de la fenêtre, l'affichage **passe** d'un **mode** à l'autre : fenêtre flottante, ancrée ou masquée.
Les fenêtres masquées sont **ouvertes** en cliquant sur l'onglet, et **fermées** en cliquant en dehors de la fenêtre.



3.9.1. Partage de la fenêtre de travail



La disposition de la fenêtre de travail peut être modifiée comme suit :

Agrandir au maximum la fenêtre (plein écran)
(changement de couleur en fonction de l'affichage en ligne/hors ligne)

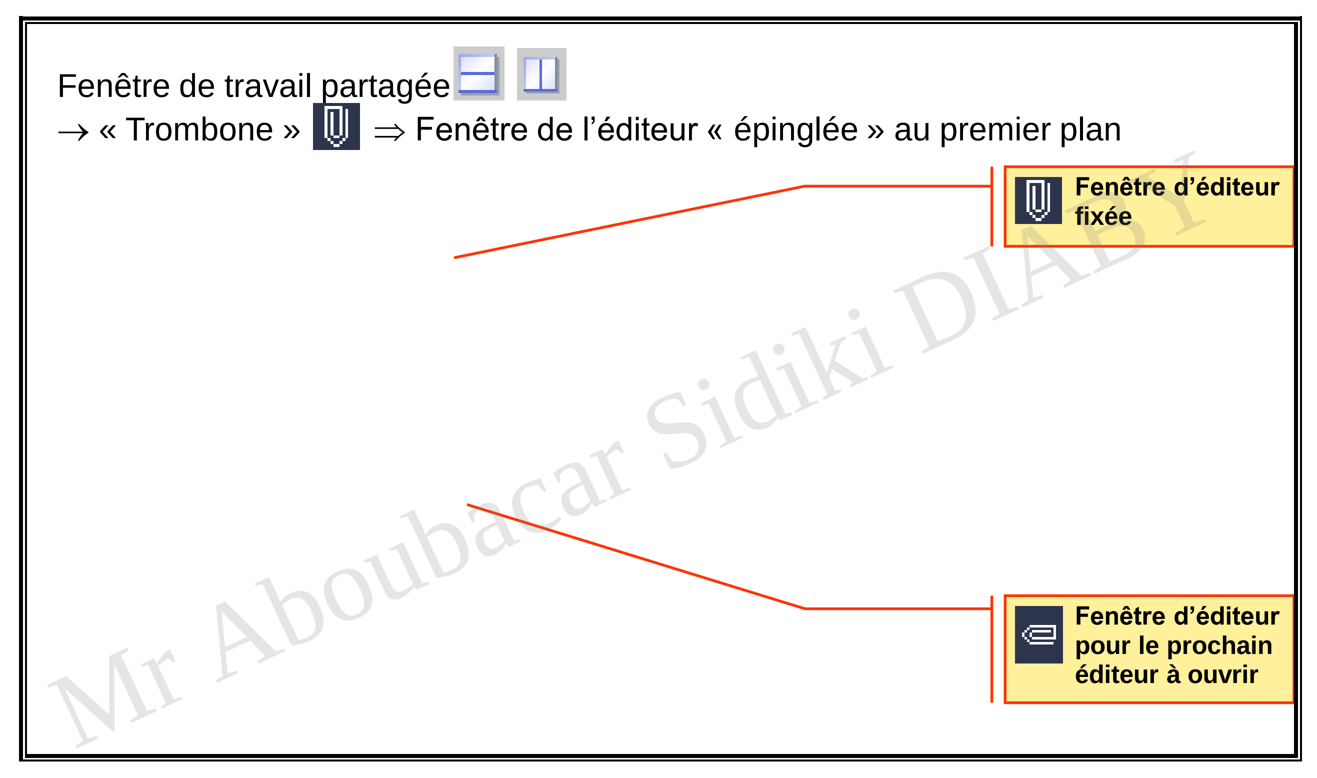
Détacher ou retirer une fenêtre de l'espace de travail
(changement de couleur en fonction de l'affichage en ligne/hors ligne)

Réintégrer une fenêtre dans l'espace de travail
(changement de couleur en fonction de l'affichage en ligne/hors ligne)

Fractionner la fenêtre dans le sens horizontal

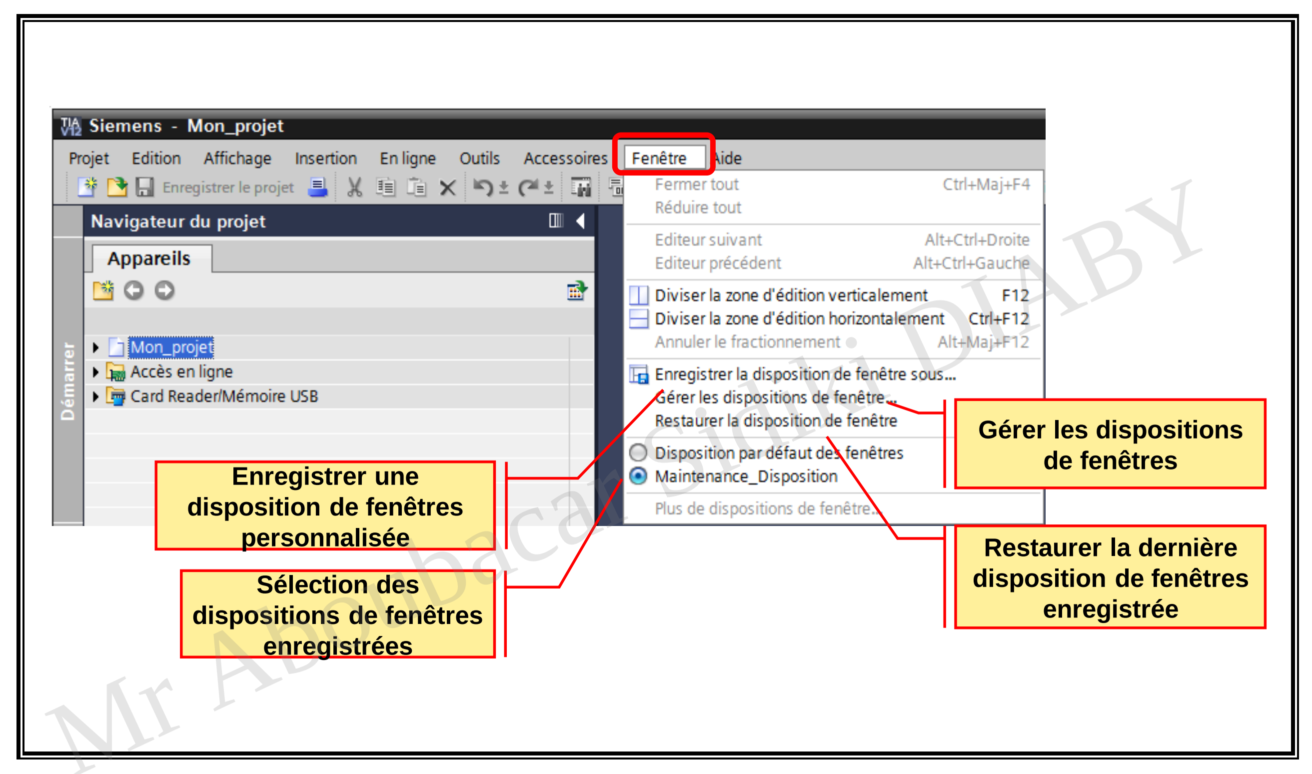
Fractionner la fenêtre dans le sens vertical

3.9.2. Maintenir l'éditeur au premier plan avec une fenêtre de travail fractionnée



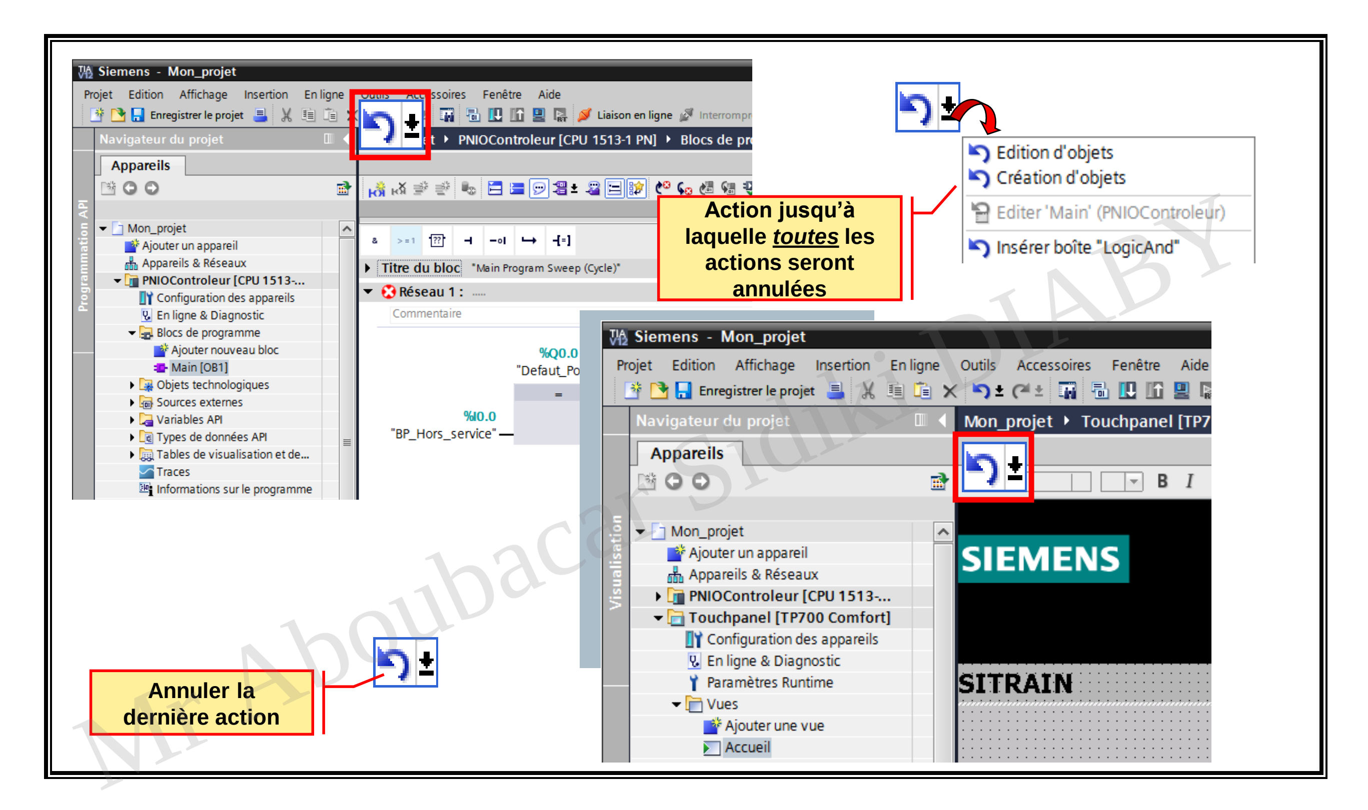
Lorsqu'on travaille avec une fenêtre de travail fractionnée, on peut « épingler » l'une des deux fenêtres de travail en cliquant sur l'icône « Trombone ». Lors de l'ouverture d'un autre éditeur, l'éditeur épinglé restera ainsi toujours au premier plan et l'éditeur nouvellement ouvert toujours au deuxième plan.

3.9.3. Enregistrer/gérer/utiliser la disposition des fenêtres



Le système d'ingénierie permet d'enregistrer et d'utiliser différentes dispositions de l'interface utilisateur.

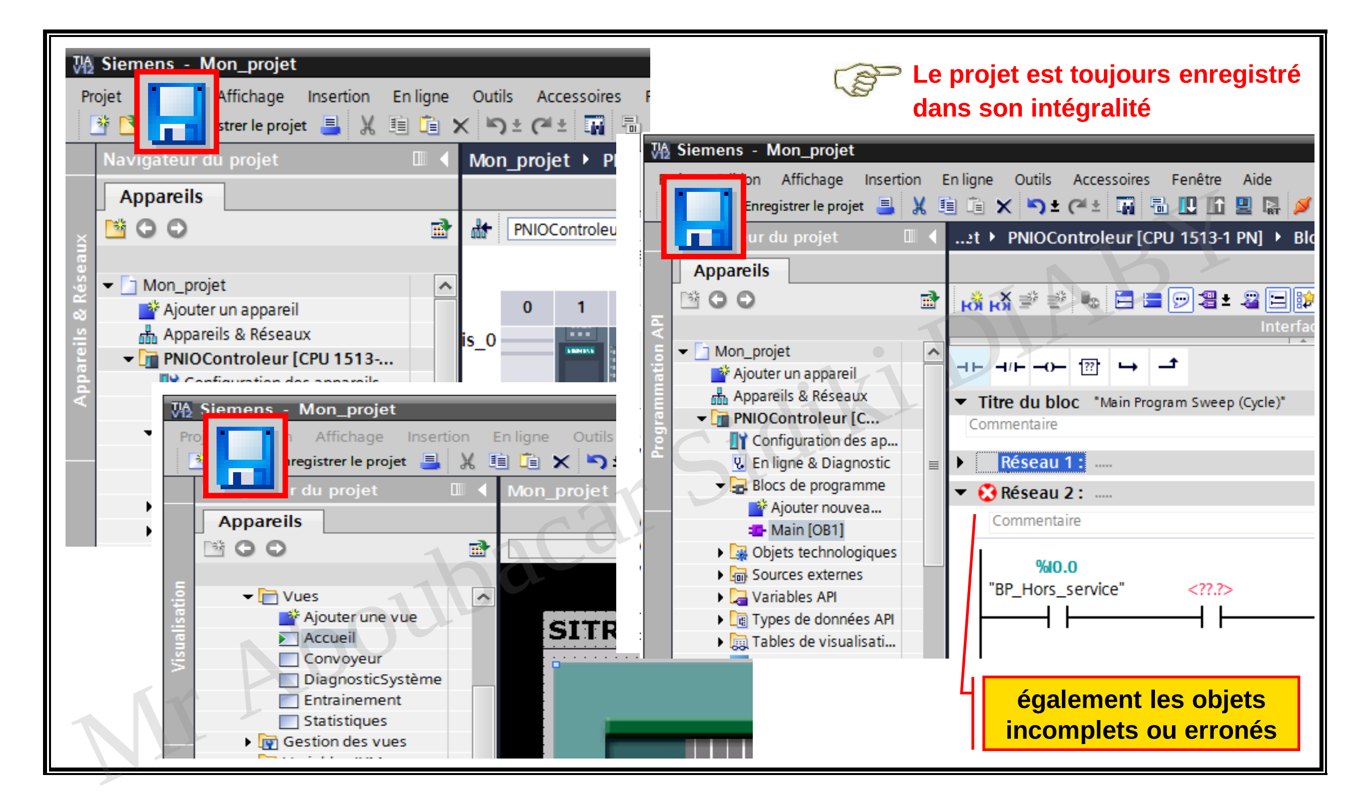
3.10. Annuler / rétablir la dernière action



Annuler les actions exécutées dans TIA Portal

Un menu déroulant permet de visualiser l'éditeur dans lequel l'annulation de la dernière action est réalisée. Les éditeurs refermés sont rouverts automatiquement. L'annulation de toutes les actions exécutées jusqu'à l'action sélectionnée permet de garantir la cohérence du projet.

3.11. Enregistrer un projet

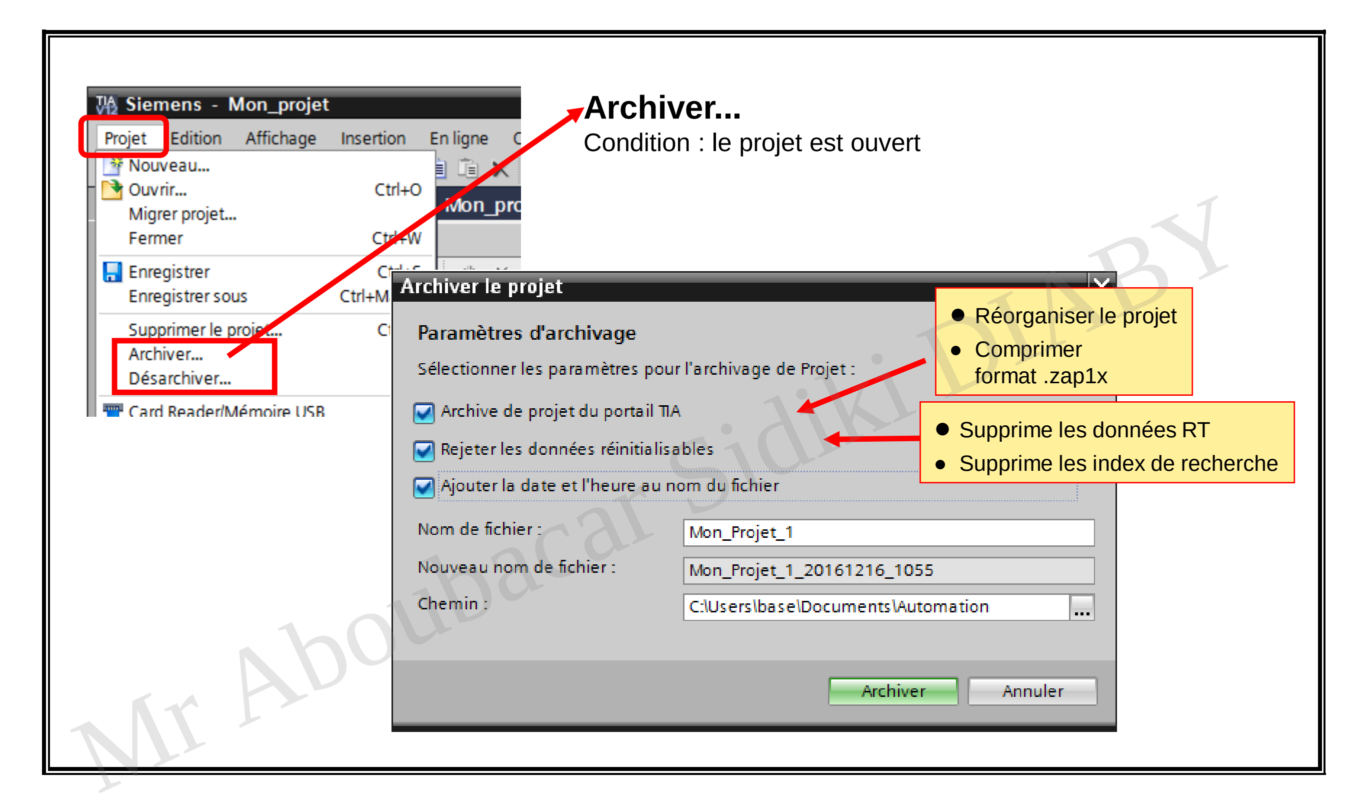


Enregistrer

Indépendamment de l'objet qui est ouvert dans la fenêtre de travail, c'est **toujours le projet complet** dans l'état actuel qui est sauvegardé par appui du bouton « Enregistrer », même si des objets du projet sont incomplets ou erronés ou incorrects (par exemple : mauvaise syntaxe dans un bloc ou mnémoniques qui ne sont pas encore assignés aux opérandes absolus dans la table de variables).

Si le projet est fermé sans sauvegarde, les modifications et les objets créés lors de la session sont perdus

3.11.1. Archiver/Désarchiver des projets



Archiver

Un projet peut être archivé dans son état actuel à n'importe quel moment.

Lorsque vous archivez le projet, vous pouvez retenir les options suivantes:

- Archive du projet TIA Portal
  Le projet est compressé (les fichiers sont réduits aux composants essentiels) et sauvegardés compressés dans une archive de projet (fichier de terminant en zap1x).
  Si ce choix n'est pas retenu, le projet est sauvegardé sous son nom et chemin comme un projet indépendant. (Enregistre sous … sans fermeture du projet courant)
- Rejeter les données réinitialisables
  Les index de recherche et les résultats de compilation HMI ne sont pas archivés.
- Ajouter la date et l'heure au nom de fichier
  l'horodatage est ajoutée à l'archive.

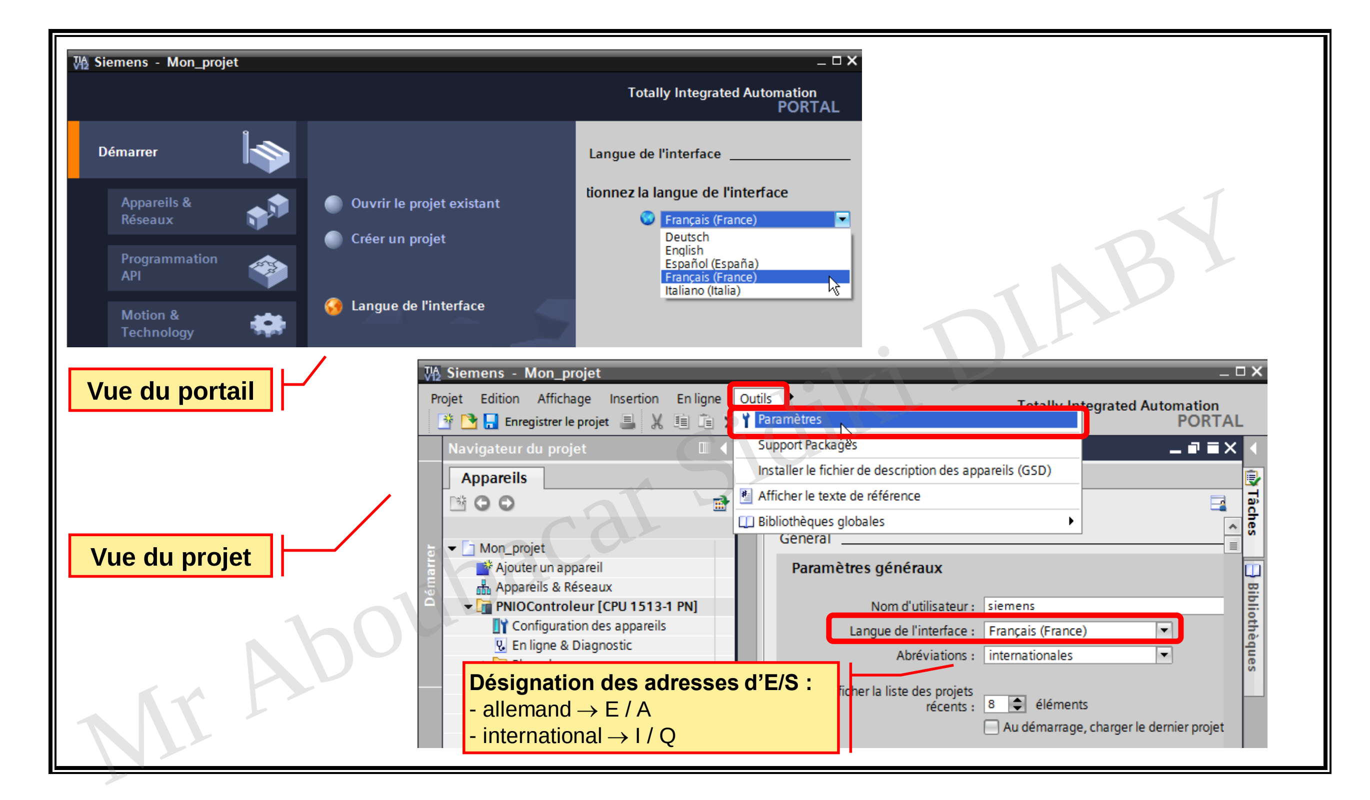
Note

La version sauvegardée du projet la plus récente est archivée. Si les dernières modifications du projet doivent y figurer, il faut au préalable sauvegarder le projet avant l'archivage.

Désarchiver

Seules des archives de projet (fichier se terminant en *zap1x*) peuvent être désarchivées et décompressées.

3.12. Langue de l'interface du système d'ingénierie

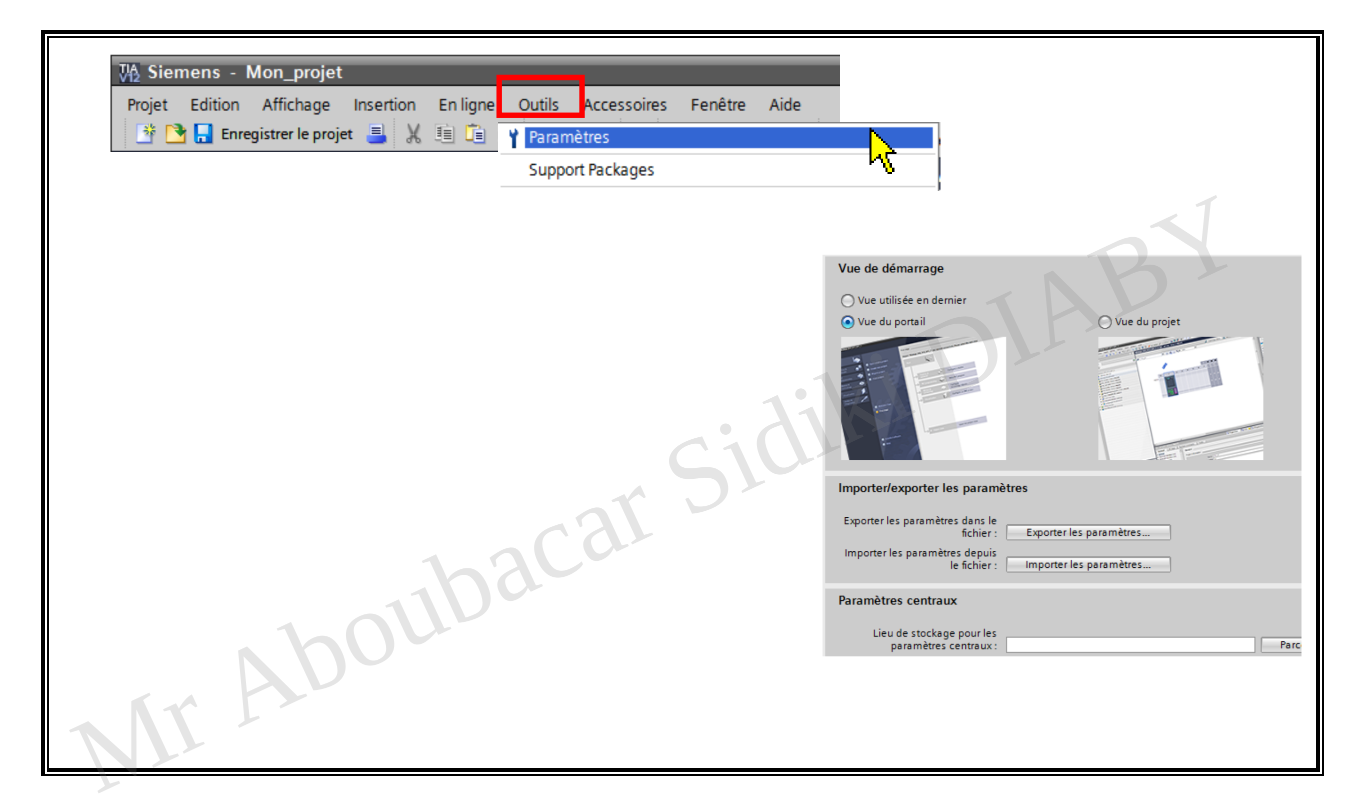


Langues disponibles au niveau de l'interface

La langue de l'interface utilisateur de TIA Portal peut être modifiée directement. Les langues suivantes sont disponibles:

- Allemand
- Anglais
- Français
- Espagnol
- Italien
- Russe
- Coréen
- Japonais
- Chinois (simplifié)

3.12.1. Paramètres : Langue, lieu de stockage, disposition



Langue

La langue de l'interface du TIA Portal peut être modifiée à tout moment sans avoir à redémarrer le système. Le TIA Portal démarre toujours dans la dernière langue sélectionnée.

Configuration de la mémoire

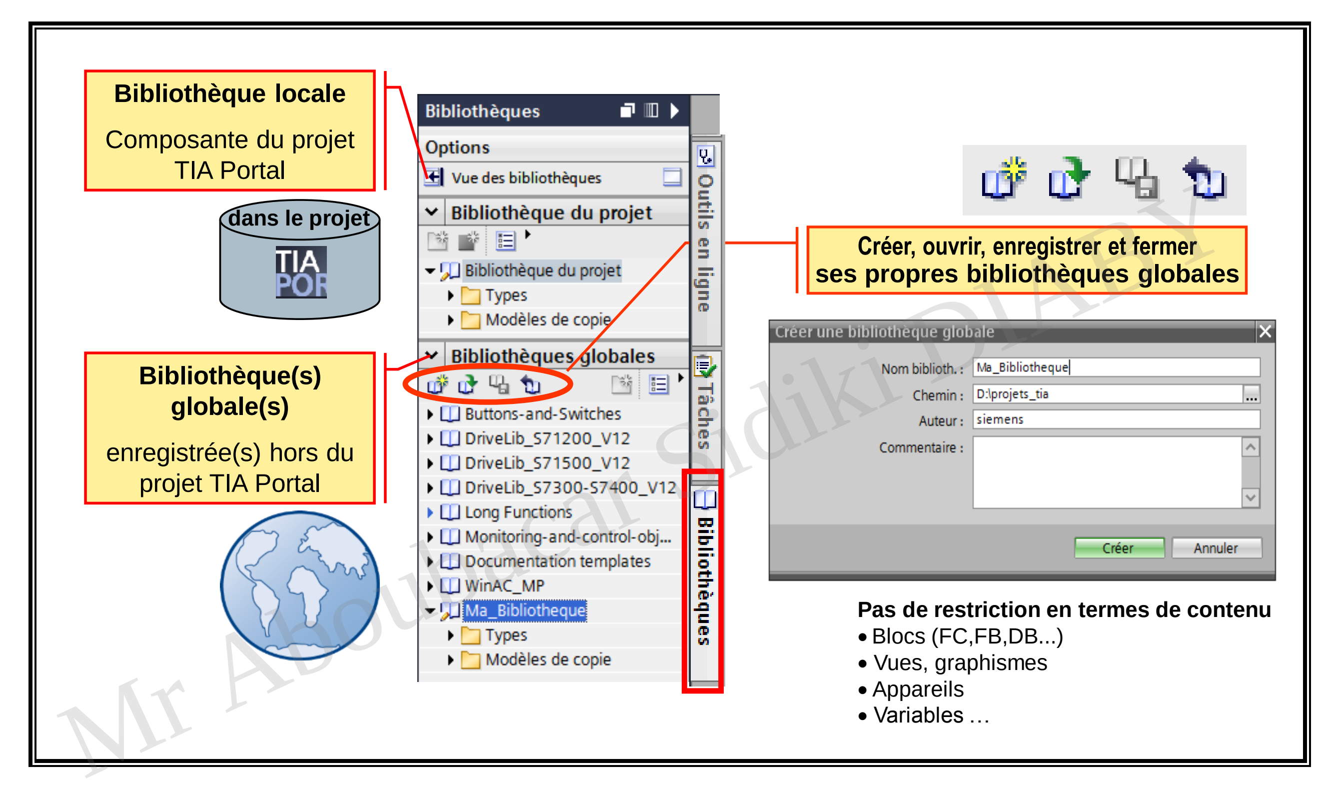
Lieu de stockage des projets :
Lieu de stockage pour les projets nouvellement créés et les bibliothèques du projet

Lieu de stockage des bibliothèques :
Lieu de stockage pour les bibliothèques globales

Disposition

La réinitialisation aux valeurs par défaut permet de restaurer la disposition d'origine de la fenêtre du TIA Portal.

3.13. Task Card « Bibliothèques »



Une bibliothèque permet de regrouper différents objets du TIA Portal. Il convient de distinguer les bibliothèques globales des bibliothèques locales liées à un projet.

Objets contenus dans les bibliothèques

Fonctions (FC), blocs de fonction (FB), blocs de données (DB), appareils, types de données API, tables de visualisation, vues, graphismes, blocs d'affichage, tables de variables...

Types d'utilisation des bibliothèques globales

Les objets contenus dans les bibliothèques peuvent servir de modèle de copie ou d'instance.

- Les objets dans le dossier **Modèles de copie** sont *copiés dans le projet lors de leur utilisation*. Les éventuelles modifications ultérieures du modèle ne sont pas reprises dans les copies utilisées dans le projet.
  → Objets au choix
- Les objets du dossier **Types** sont *copiés dans le dossier Types de la bibliothèque de projet lors de leur utilisation et une instance (occurrence) est créée dans le projet.*
  Ces objets sont enregistrés ensuite dans la bibliothèque locale de projet, y sont traités et le projet ne contient qu'une occurrence sous forme de référence qui renvoie vers l'objet central. Si des objets centralisés de ce type sont modifiés, toutes les instances (occurrences) sont mises à jour dans le projet.
  → uniquement blocs d'affichage et structures dans WinCC

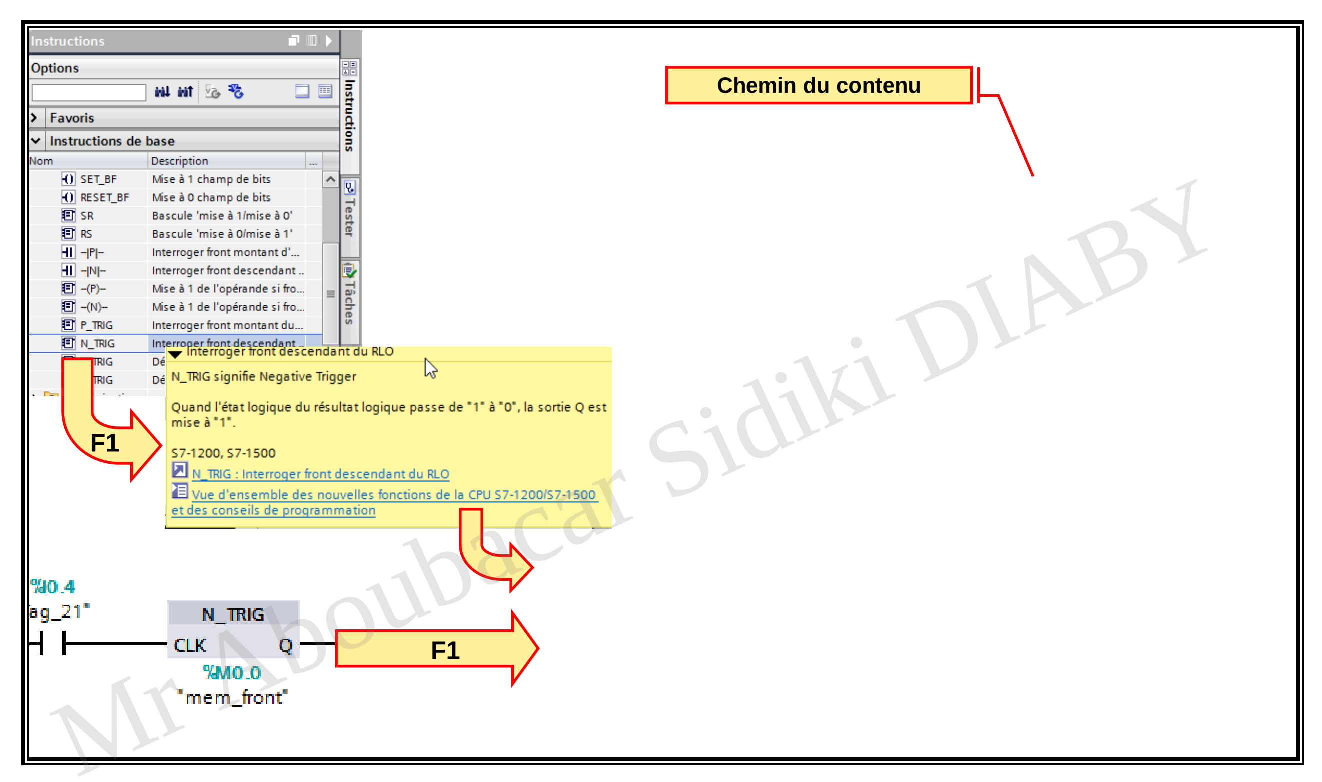
Bibliothèque du projet

Chaque projet possède sa propre bibliothèque. Celle-ci contient les objets qui seront réutilisés au sein du projet. La bibliothèque du projet est systématiquement ouverte, enregistrée et fermée en même temps que le projet en cours de traitement

Bibliothèques globales

Les bibliothèques globales sont sauvegardées indépendamment des projets et servent à enregistrer des objets susceptibles d'être réutilisés dans plusieurs projets. Les bibliothèques globales comportent également des bibliothèques fournies du TIA Portal. Les bibliothèques fournies ne peuvent pas être modifiées.

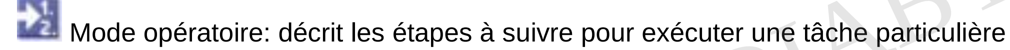
3.14. Fonctions d'aide



Des fonctions d'aide complètes sont disponibles pour résoudre vos tâches; elles décrivent les concepts de base, les actions et les fonctions.

- **Info-bulle** pour information au niveau des éléments de l'interface utilisateur, par exemple pour les instructions, les zones de saisie, les boutons et les icônes.

Quelques Info-bulles sont cascadées avec des informations plus précises.


Mode opératoire: décrit les étapes à suivre pour exécuter une tâche particulière

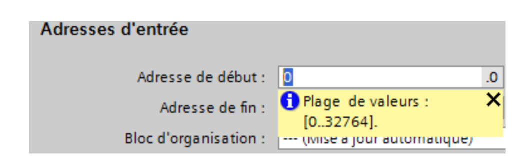

Exemple: Application expliquant la résolution d'une tâche d'automatisation.

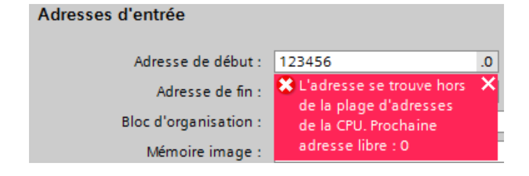
Informations: Informations contextuelles pour les fonctions de TIA Portal


Références: Information détaillée au sujet des instructions et objets

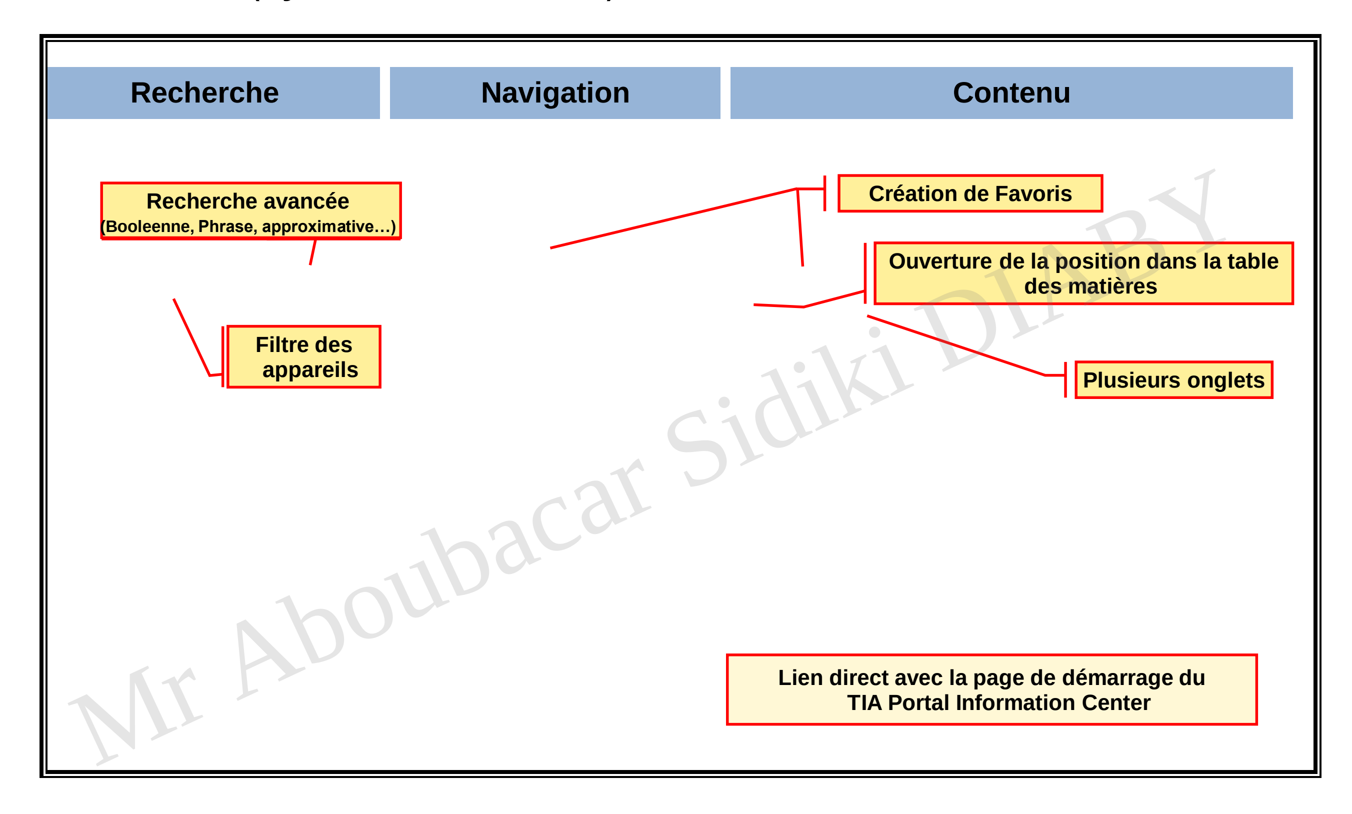
Ils sont activés par clic (ouverture de l'aide)

- Aide contextuelle
  Par exemple, au niveau des options du menu par appui sur la touche <F1>.
- Dans les boites de saisie (par exemple, dans les propriétés de la fenêtre d'inspection), le menu déroulant fournit les informations sur les plages de valeurs et les type de données admissibles.





3.14.1. Aide (Système d'information)

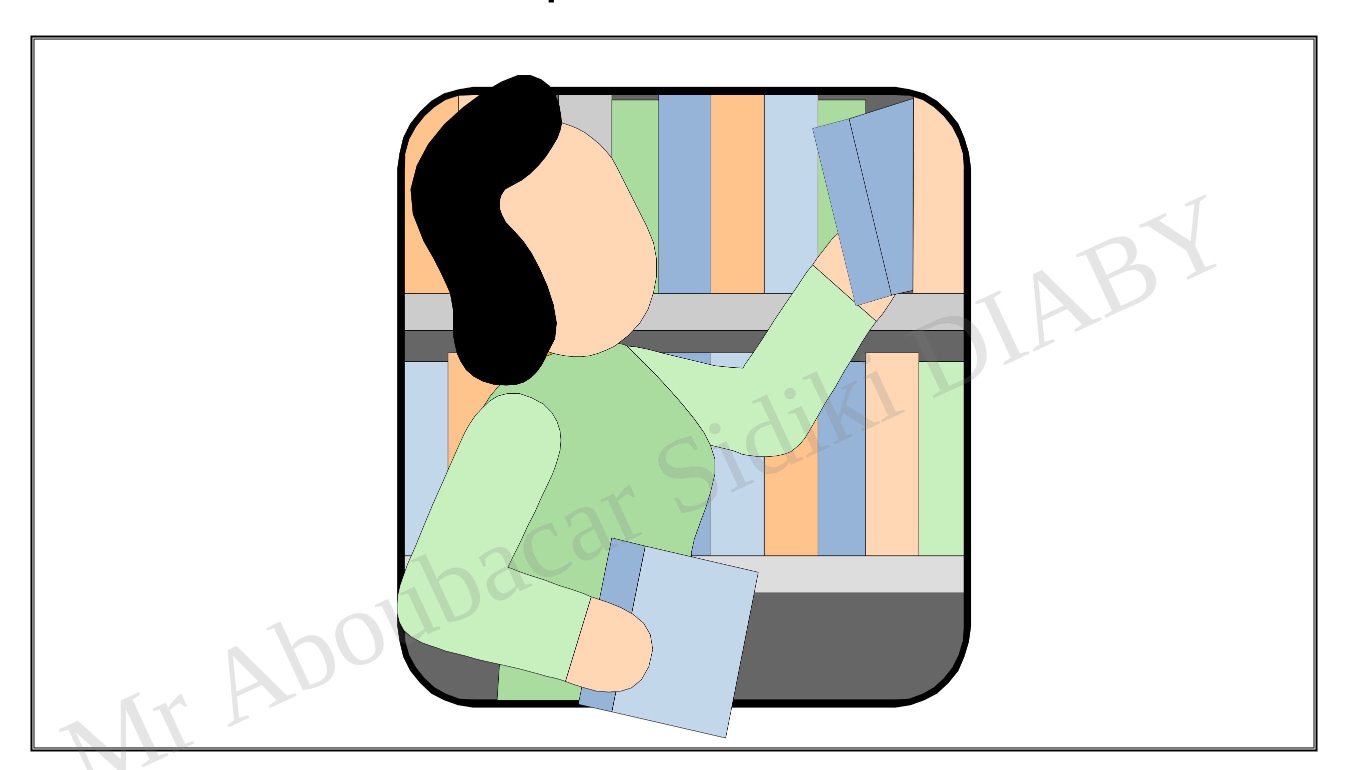


Contenu du système d'information

- Zone de recherche
  Dans la zone de recherche vous pouvez lancer une recherche d'un texte complet dans l'ensemble des sujets de l'aide.
- Zone de navigation
  Dans la zone de navigation vous trouvez la table des matières ainsi que les favoris.
- Zone du contenu
  Les pages d'aide sont affichées dans cette zone du contenu. Vous pouvez ouvrir différents onglets afin d'afficher simultanément plusieurs pages d'aide.

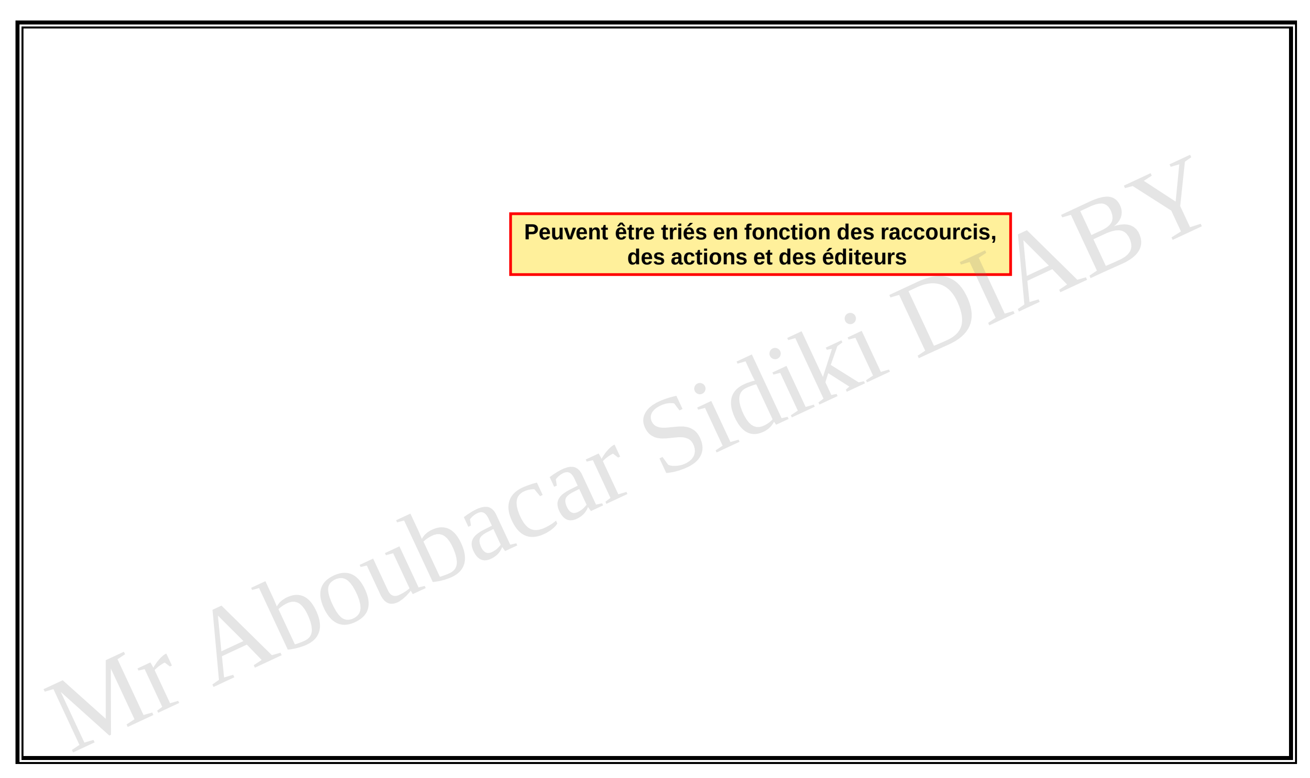
Vous pouvez afficher ou masquer les zones individuellement en utilisant les flèches du séparateur de fenêtres. Ainsi vous fermez la zone de recherche et de navigation pour augmenter, si nécessaire la taille de la zone du contenu.

3.15. Informations complémentaires



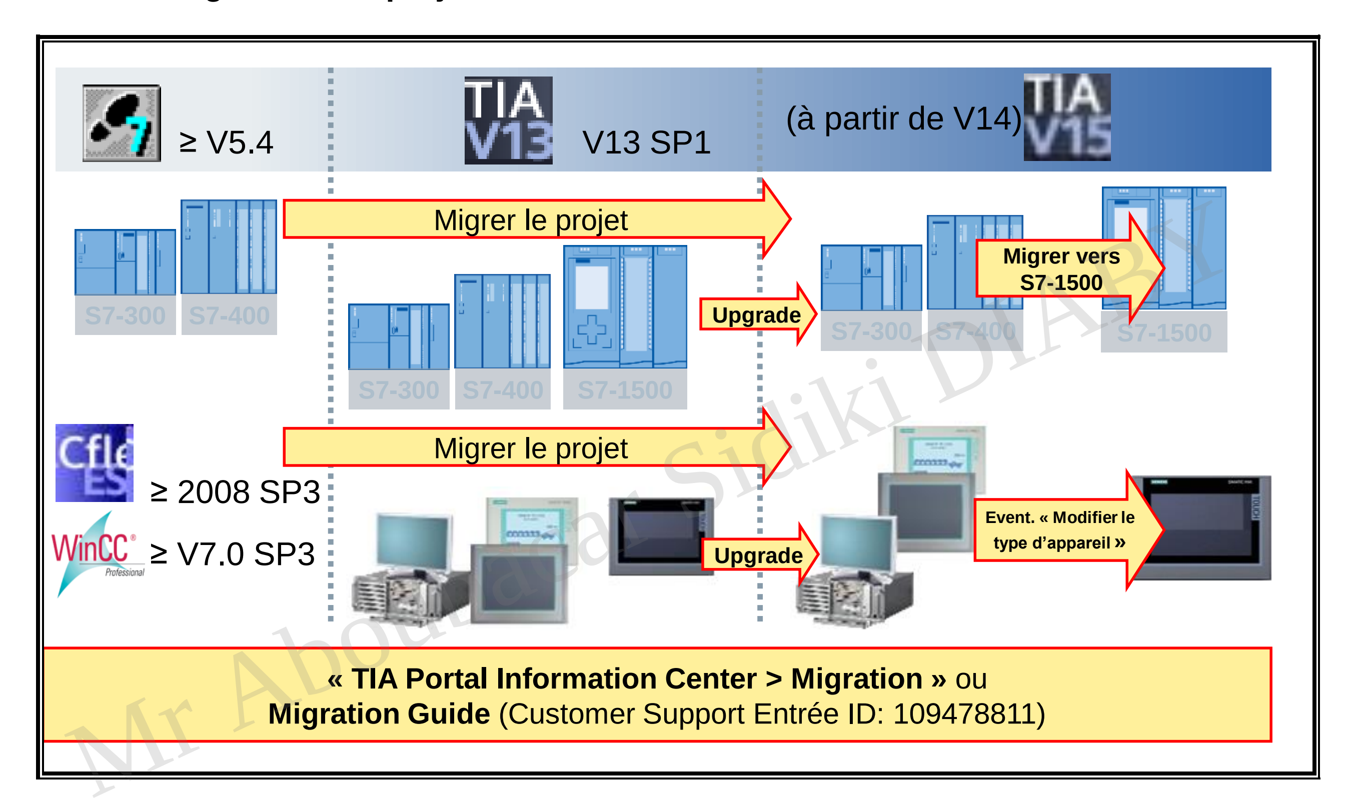


3.15.1. Raccourcis clavier dans le TIA Portal



Des raccourcis clavier sont disponibles dans le système d’ingénierie pour les fonctions importantes et fréquemment utilisées.

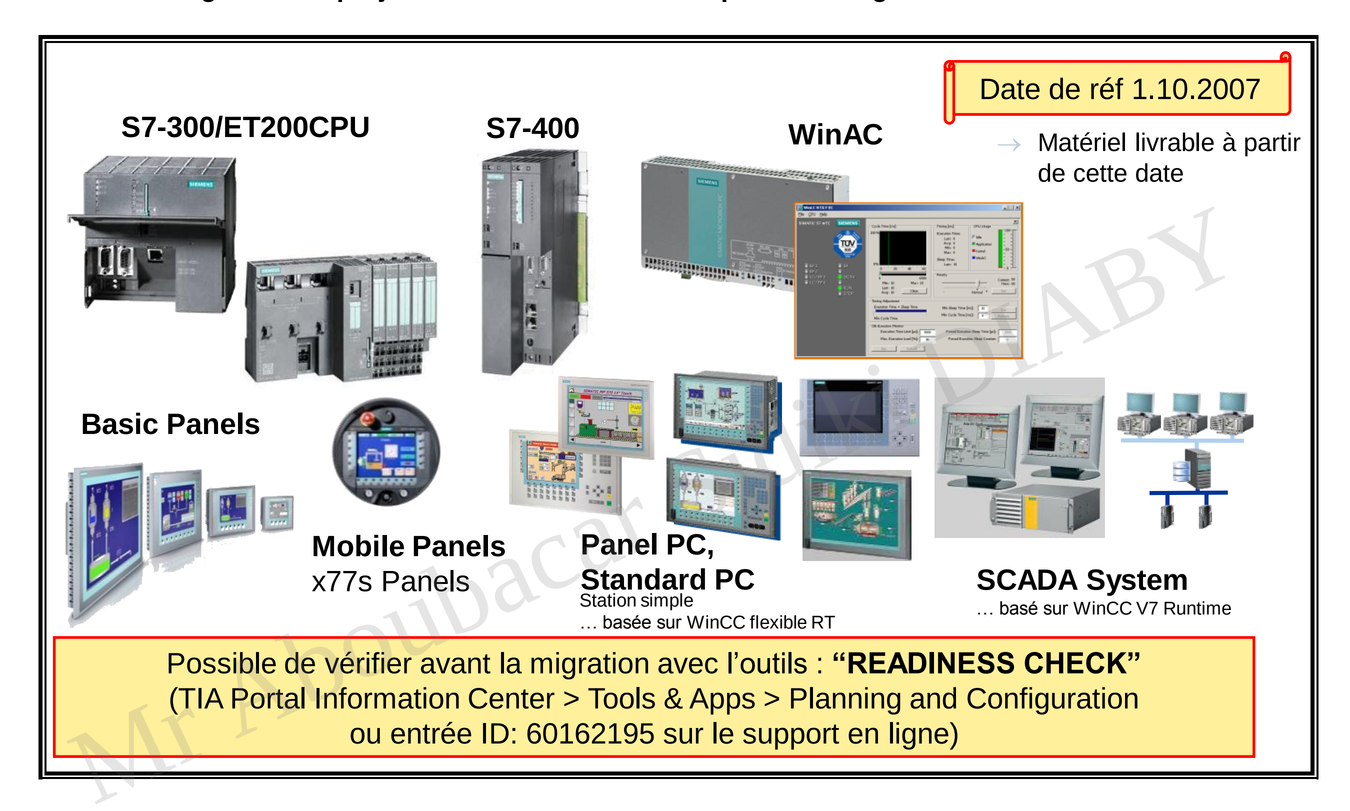
3.15.2. Migration des projets



Pour pouvoir être migrés, les projets issus d'anciennes versions STEP7, WinCC flexible ou WinCC doivent être convertis au préalable dans la version STEP7/WinCC flexible/WinCC indiquée ci-dessus.

La migration de programmes S7-300/400 vers le matériel S7-1500 est possible au sein de V1x.

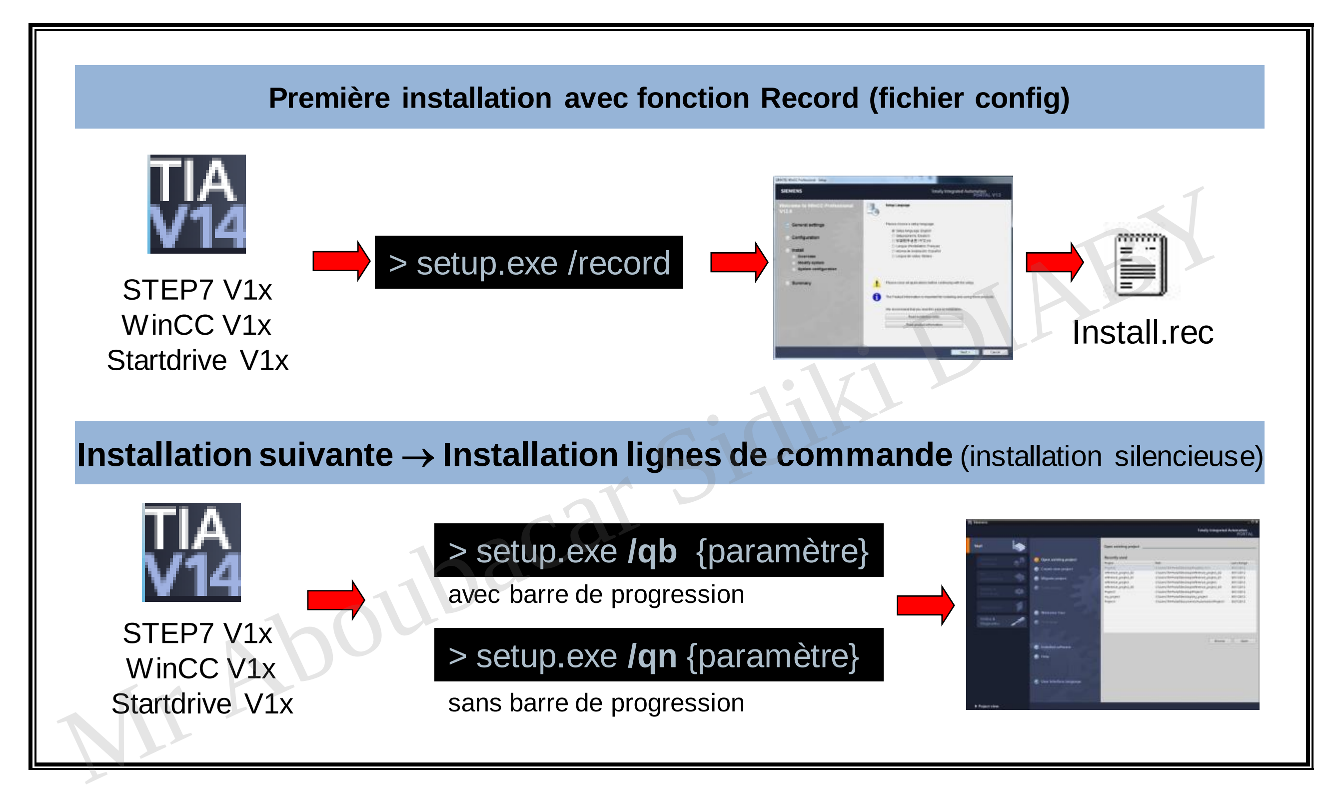
3.15.2.1. Migration de projets STEP7 V 5.x : matériel pris en charge



A ce niveau, la configuration matérielle peut également être migrée. Dans l'autre cas seul le programme peut être migré et adapté au hardware supporté.

Il est possible de vérifier en amont si le matériel est supporté dans TIA Portal. Pour cela vous pouvez utiliser l'outil « READINESS CHECK » téléchargeable au niveau du Support (https://support.industry.siemens.com) dans l'article ID: **60162195** ou via le «TIA Portal Information Center ».

3.15.3. Multi-installation → Installation avec fonction Record dans le Setup

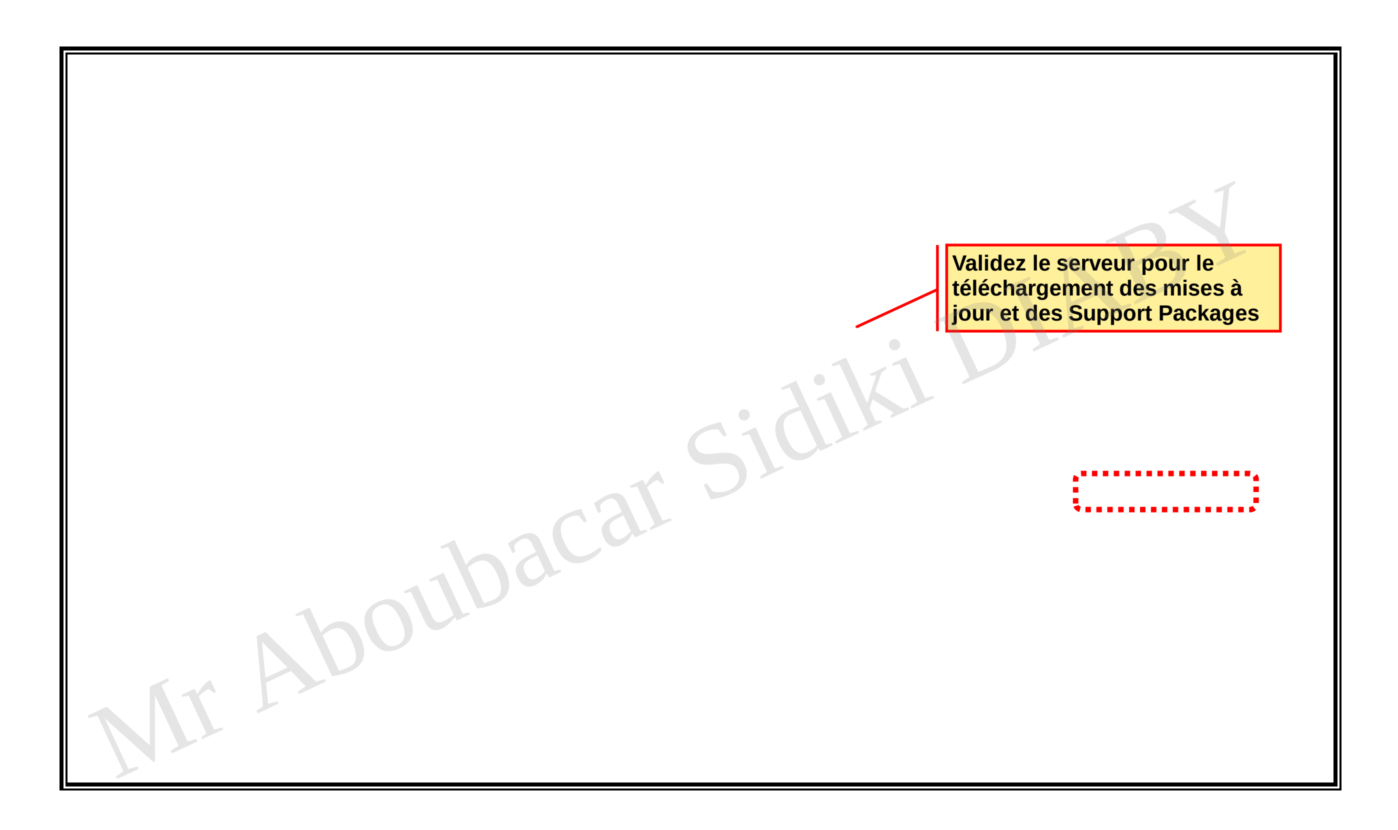


Lorsque plusieurs installations personnalisées identiques sont requises, il est possible d'enregistrer la première installation  Install.rec, puis de reproduire cette dernière sous forme d'installations silencieuses.

Installation silencieuse

Les installations suivantes s'effectuent sans interrogations de manière identique à la première installation.

3.15.4. Mise à jour logiciel : Automation Software Updater



Mise à jour logiciel :

Le dialogue « Mises à jour disponibles » vous permettra de rechercher les mises à jour.

- Lancement via: Vue Portail: Démarrer > Logiciels installés > Rechercher des mises à jour
  Vue du projet: Aide > Logiciels installés > Rechercher des mises à jour
- Vérifie et recherches les mises à jour disponibles pour les logiciels installés
- Téléchargement des mises à jour
- Interrompre / poursuivre les téléchargements
- Installer les mises à jour.

Sommaire

4

4. Appareils & Réseaux, Configuration du matériel et outils en ligne

A l'issue du chapitre, vous allez ...



... savoir établir une liaison en ligne via un réseau Industrial Ethernet entre une console de programmation (PG) et la CPU d'un automate

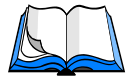
… savoir arrêter, démarrer, réinitialiser avec les valeurs par défaut la CPU avec les fonctions en ligne

... savoir créér une nouvelle station et assigner les paramètres

... savoir créer et paramétrer une configuration prédéfinie.

... savoir déterminer et définir l'adressage des modules d'entrée/sortie d'un automate S7-1500

... savoir charger une configuration matérielle et le programme

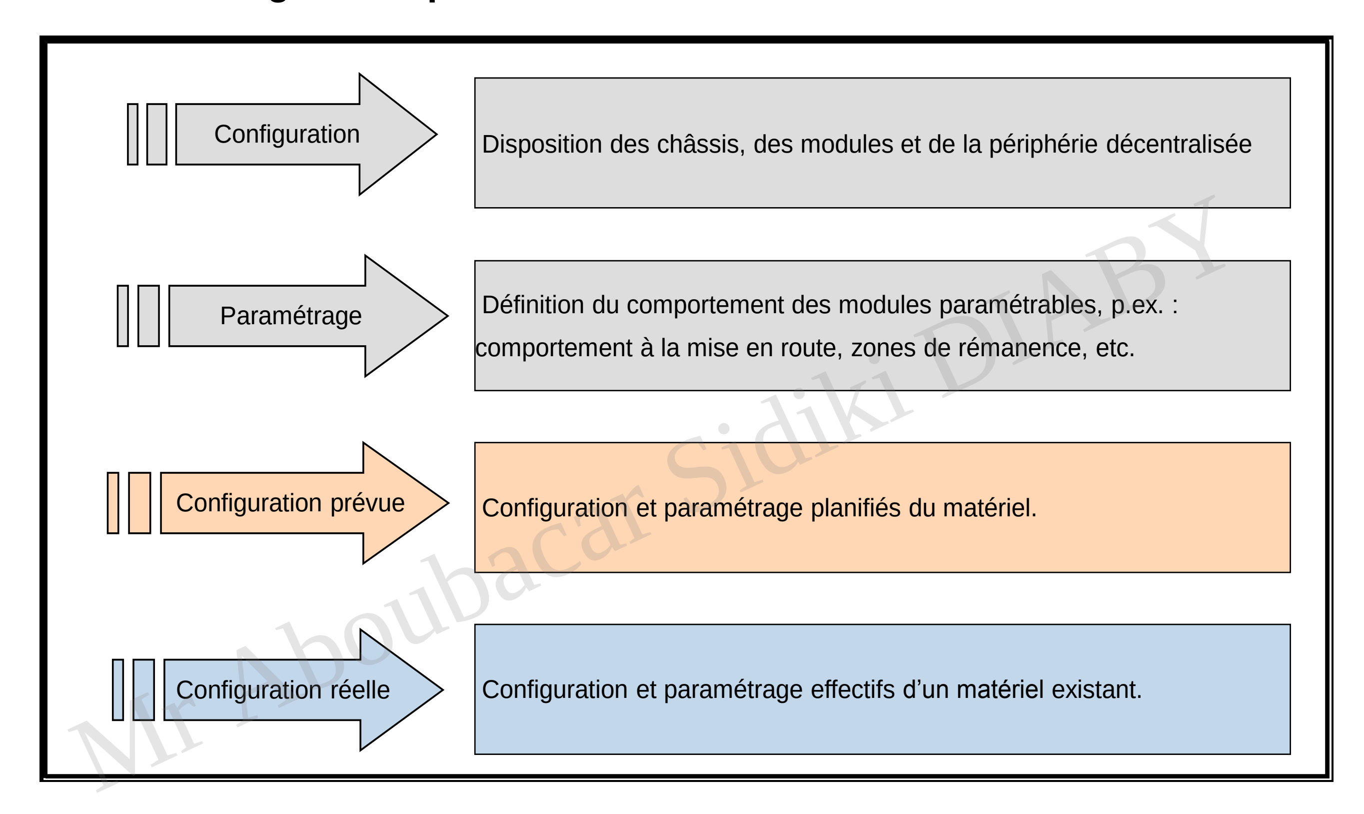


Ce chapitre présente la configuration matérielle d'un système d'automatisation S7-1500.

A l'issue de ce chapitre, vous saurez configurer le matériel utilisé pour la formation. Vous aurez une vue d'ensemble du diagnostic matériel des modules (LED), des fonctions en ligne et de l'adressage par défaut du S7-1500.

Objectifs pédagogiques :
Savoir mettre en service un système S7-1500 doté d'une périphérie centralisée.

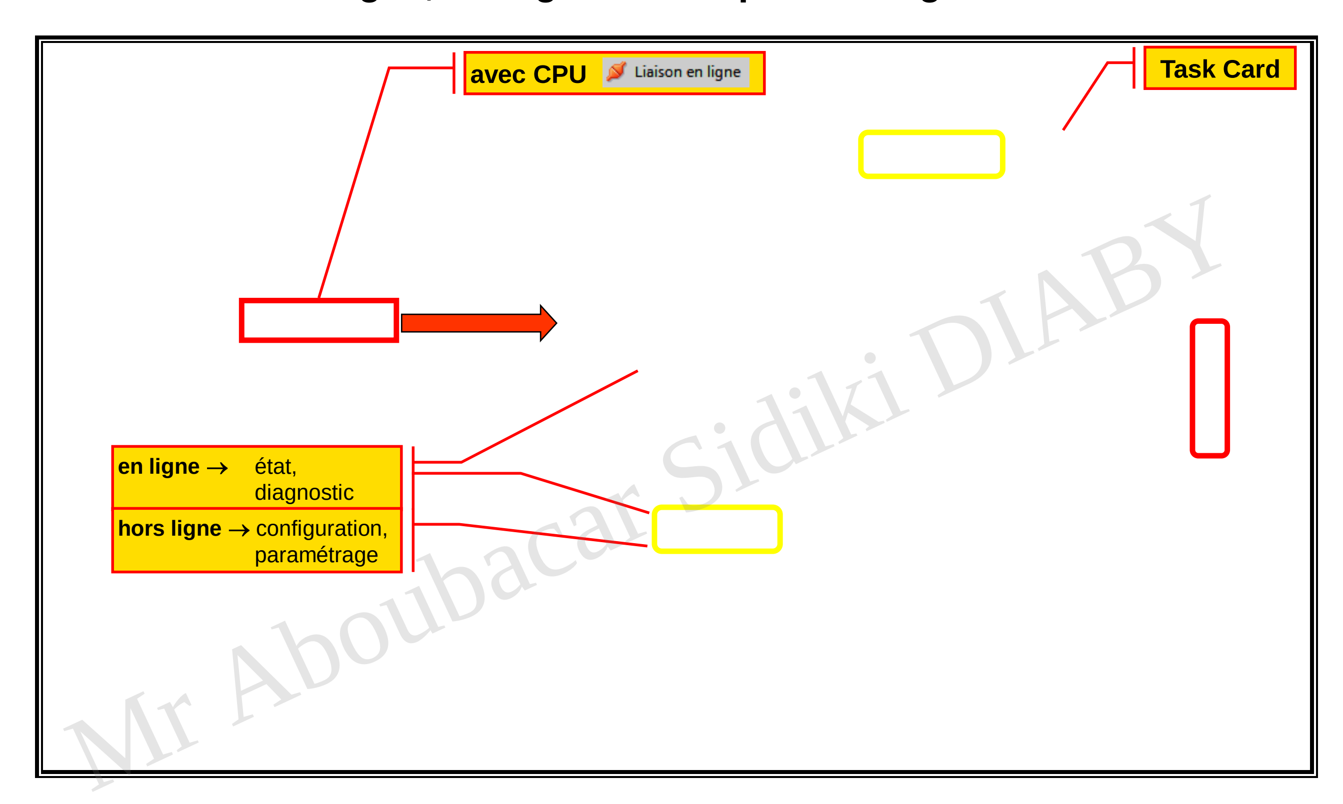
4.1. Configuration prévue et réelle



Configuration prévue et réelle

Lors de la configuration d'une installation, on crée en fait une configuration personnalisée (ou configuration prévue). Celle-ci contient une station matérielle avec les modules prévus et leurs paramètres. L'automate est configuré selon ces paramètres et la configuration prévue est chargée dans la CPU lors de la mise en service.

4.2. Outils en ligne, configuration et paramétrage du matériel



Outils en ligne

Si une liaison en ligne peut être établie avec la CPU, vous pouvez appeler des informations de diagnostic et d’état sur tous les modules.

Lorsqu’une CPU est accessible en ligne, vous pouvez sélectionner son mode de fonctionnement via la Task Card « Outils en ligne » et afficher d’autres informations d’état la concernant (statistiques sur le temps de cycle et le taux d’occupation des mémoires).

Configuration et paramétrage du matériel

Presque tous les appareils ou composants d’une solution d’automatisation (comme les API ou les pupitres opérateur à écran tactile) sont paramétrables. Les paramétrages nécessaires à la mise en service du système s’effectuent à l’aide de l’éditeur « Appareils & Réseaux ».

Ils consistent par exemple à affecter à tous les composants d’un réseau Ethernet les adresses IP qui leur serviront à communiquer ultérieurement entre eux.

Au sein de l’automate, ils permettent par exemple de définir les plages d’adresses des modules d’E/S et de régler le temps de surveillance de cycle de la CPU.

4.3. Liaison en ligne via Industrial Ethernet : adresse IP et masque de sous-réseau



Protocole IP

Le protocole IP (Internet Protocol) constitue la base de tous les réseaux TCP/IP. Il élabore les datagrammes (paquets de données spécialement adaptés au protocole IP) et assure leur transport au sein du sous-réseau local ou leur routage (transmission) vers d'autres sous-réseaux.

Adresses IP

Les adresses IP ne sont pas affectées à un ordinateur particulier, mais aux interfaces réseau dont est doté un ordinateur. Lorsqu'un ordinateur possède plusieurs connexions réseau (routeurs, par ex.), une adresse IP doit donc être affectée à chacune des connexions.

Les adresses IP se composent de 4 octets. En notation décimale à point, chaque octet de l'adresse IP est exprimé par un nombre décimal compris entre 0 et 255. Les quatre nombres décimaux sont séparés entre eux par des points (voir figure).

Adresse MAC

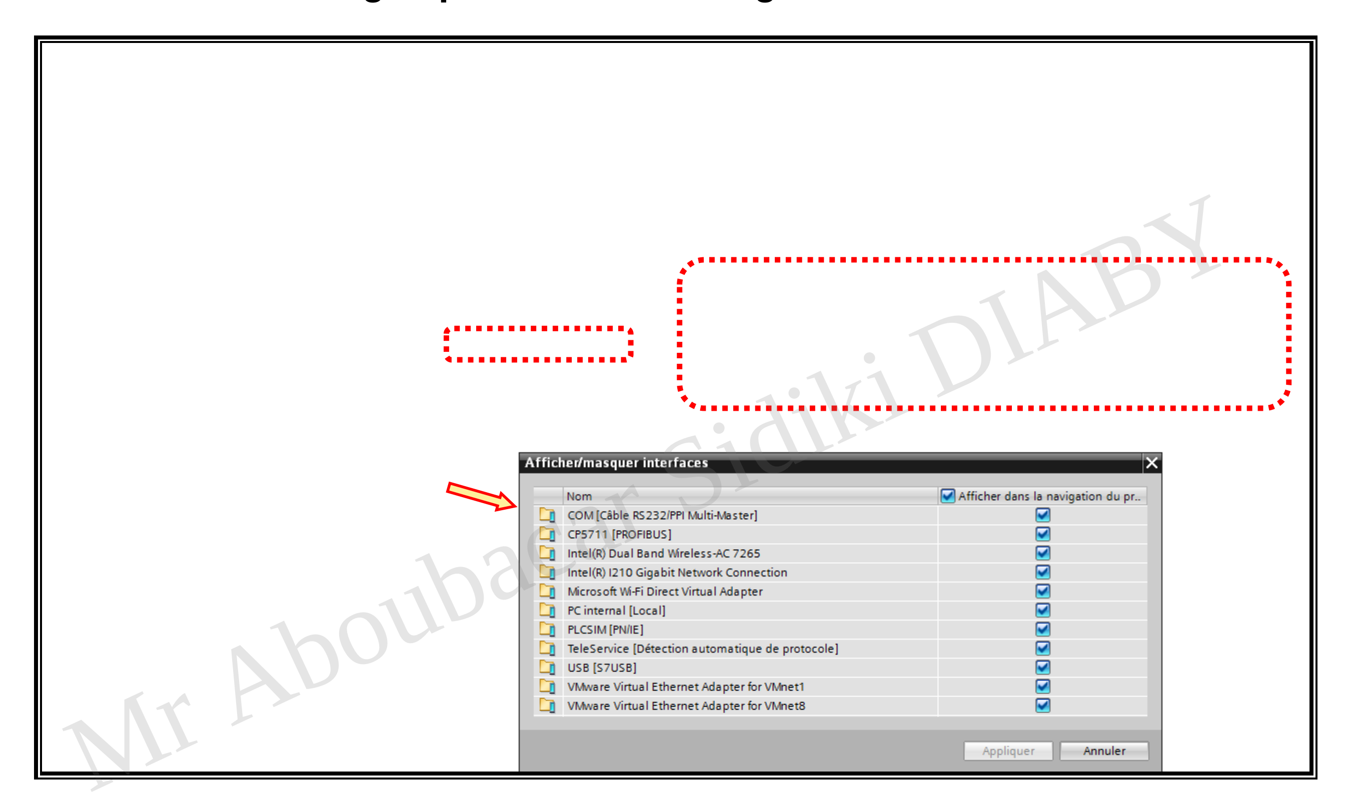
Chaque interface Ethernet a reçu de son constructeur une adresse fixe et unique au niveau mondial. Cette adresse est appelée adresse matérielle ou adresse MAC (Media Access Control). Elle est mémorisée dans la carte réseau et sert à l'identification sans équivoque de l'interface dans un réseau local. La coopération entre les constructeurs garantit une utilisation unique de chaque adresse au niveau mondial.

Masque de sous-réseau

Le masque de sous-réseau définit les adresses IP accessibles au sein du réseau. Il sépare l'adresse IP en partie réseau et en partie appareil (poste raccordé).

Seules sont accessibles les adresses IP dont la partie réseau est identique.
Par ex. : masque de sous-réseau = **255.255.255.0** et adresse IP = **192.168.111.10**
adresses IP accessibles : **192.168.111.1** à **192.168.111.254**

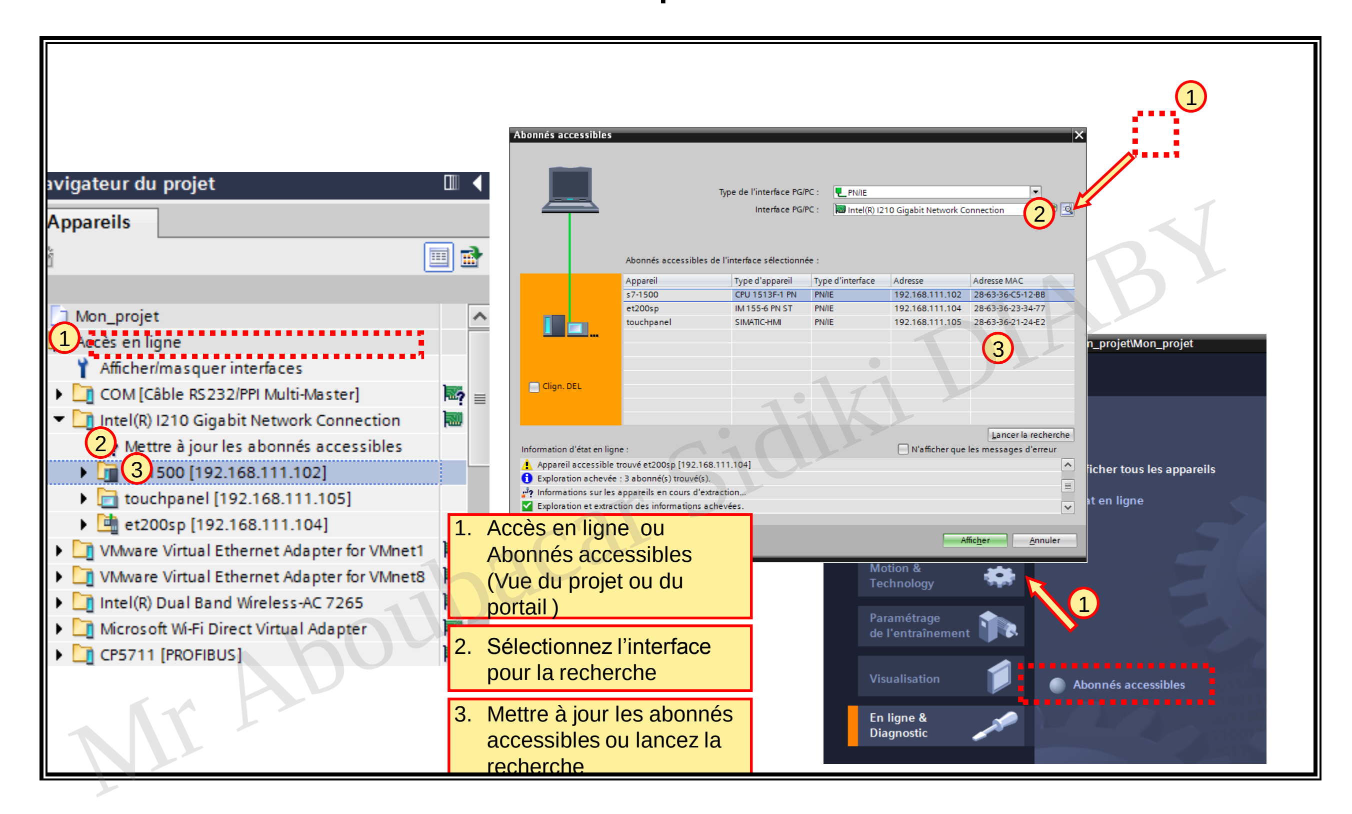
4.4. Accès en ligne par défaut et filtrage des interfaces



Au niveau des réglages vous pouvez retenir un paramétrage par défaut pour le chemin de liaison des accès en ligne.

Toutes les interfaces possibles pour la PG/PC sont affichées dans le répertoire des accès en ligne. Comme elles ne sont pas toutes nécessaires ou exploitables, on peut les masquer pour plus de visibilité.

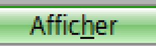
4.4.1. Abonnés accessibles → vue du portail



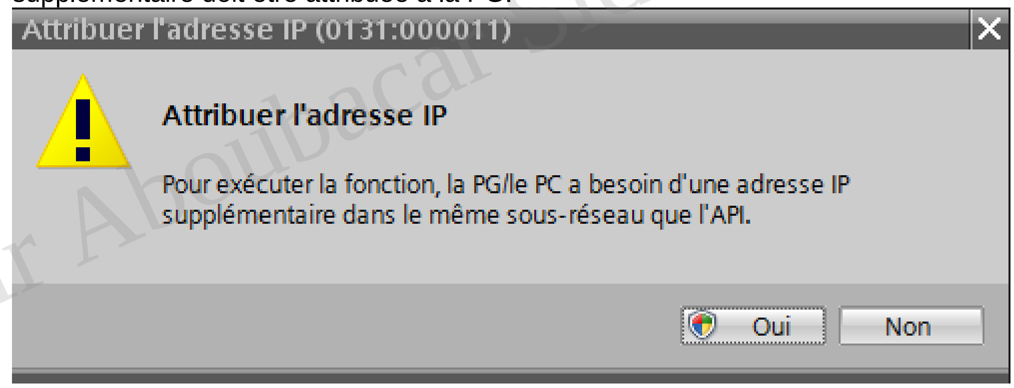
Abonnés accessibles dans la vue du portail

Cette fonction offre une possibilité d'accès rapide (par ex. à des fins de dépannage), même lorsqu'aucune donnée de projet hors ligne relative aux systèmes cibles n'est présente dans la PG.

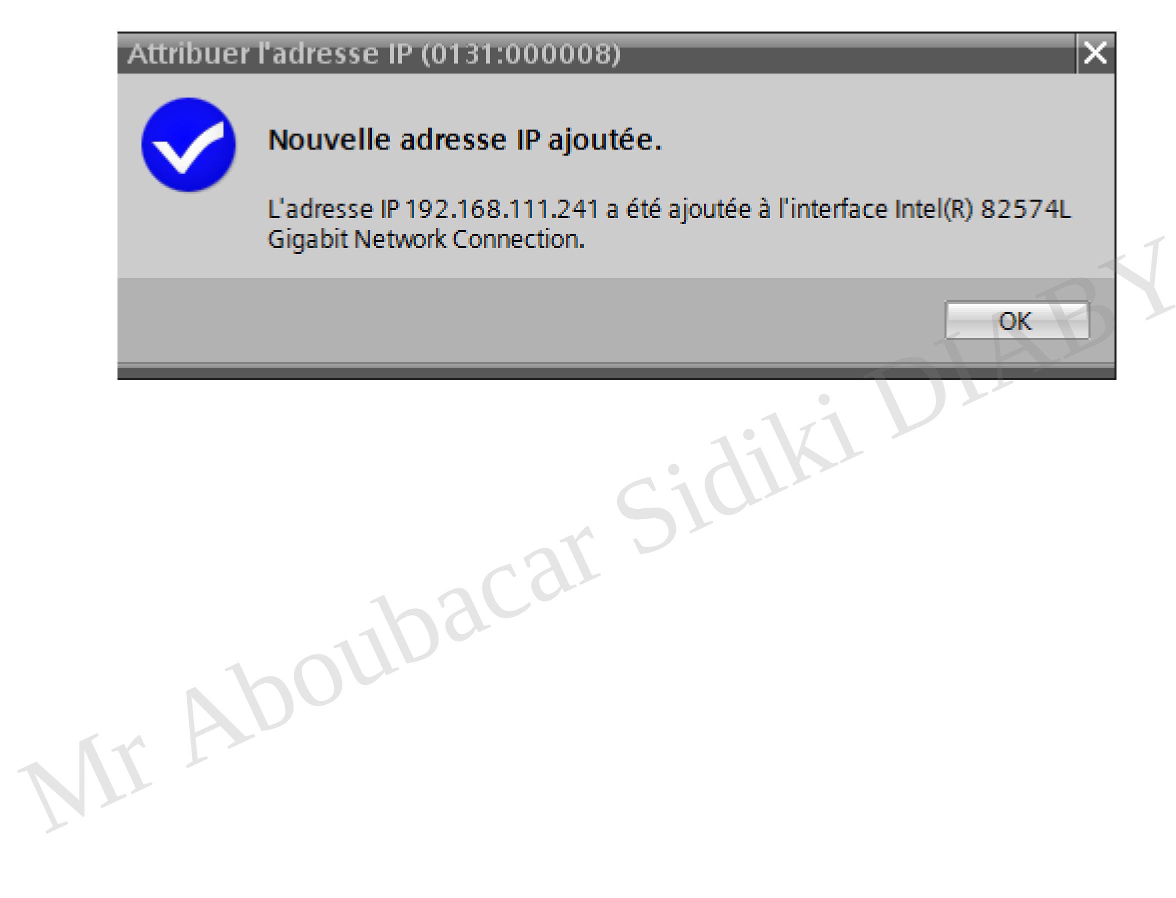
La vue du portail liste tous les modules programmables accessibles (CPU, FM, CP, pupitres opérateur), même s'ils se trouvent dans d'autres sous-réseaux.

Accéder aux fonctions en ligne → bouton 

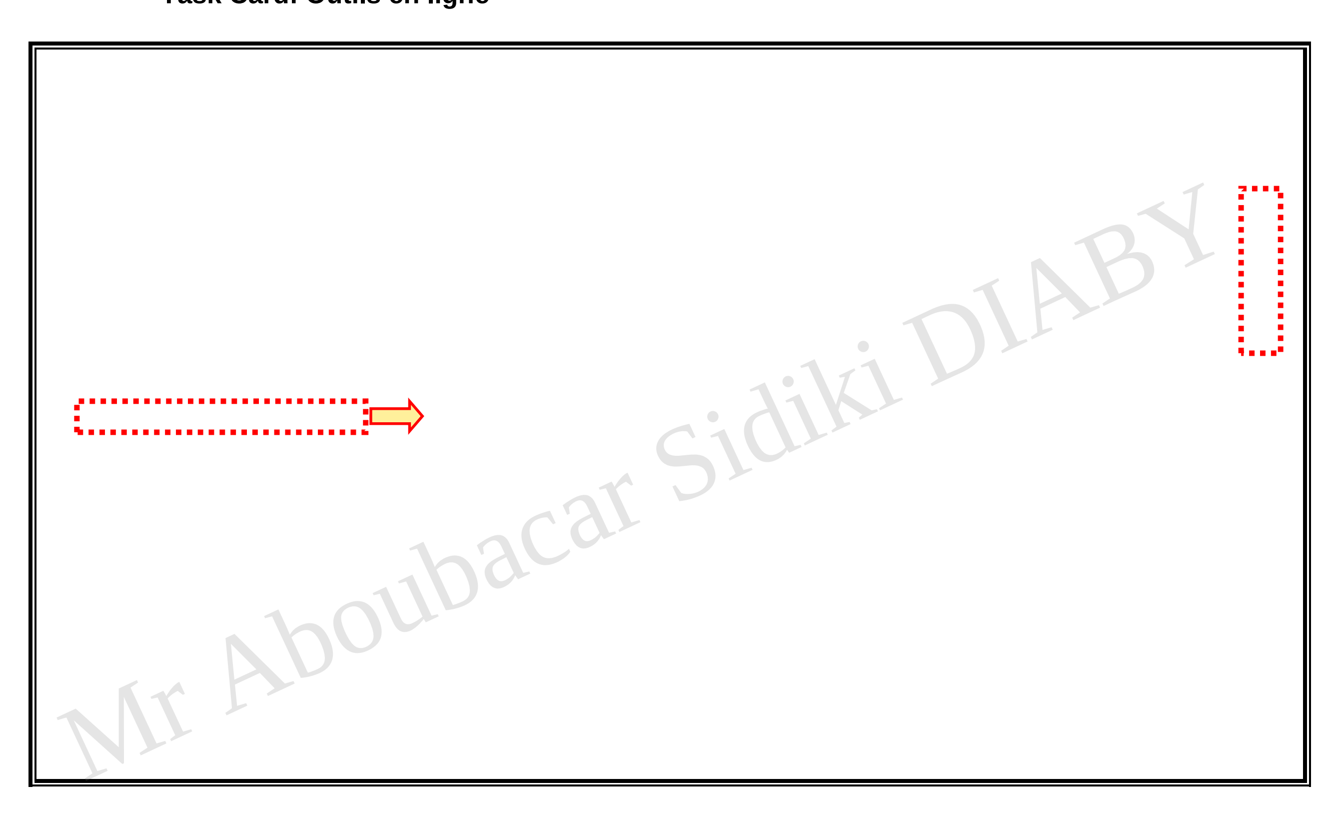
Lorsque vous cliquez sur le bouton « Afficher » pour accéder en ligne à un module situé dans un autre sous-réseau que la PG, une boîte de dialogue s'affiche pour demander si une adresse IP supplémentaire doit être attribuée à la PG.



Si vous répondez « Oui », une adresse IP supplémentaire située dans le même sous-réseau que l'adresse de la CPU est attribuée à la PG. Toutes les fonctions en ligne sont ensuite utilisables.


Attribuer l'adresse IP (0131:000008)
Nouvelle adresse IP ajoutée.
L'adresse IP 192.168.111.241 a été ajoutée à l'interface Intel(R) 82574L Gigabit Network Connection.
OK

4.4.1.1. Accès en ligne : En ligne & Diagnostic, Task Card: Outils en ligne



Panneau de commande CPU

L'état de fonctionnement de la CPU peut être modifié.

- RUN → STOP :

Lors du passage de RUN à STOP, la CPU met fin au traitement en cours du programme utilisateur.

- STOP → RUN :

Lors du passage de STOP à RUN, la CPU effectue un redémarrage.

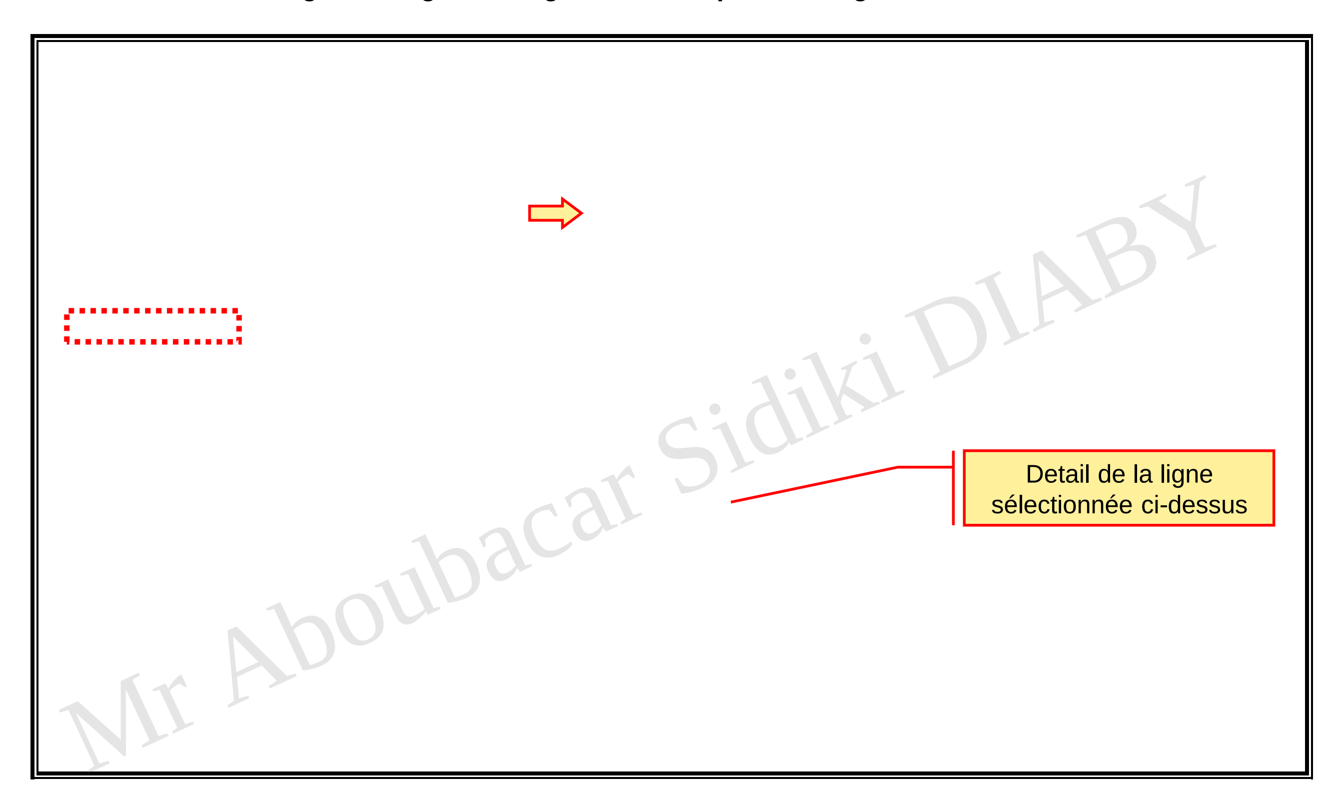
Temps de cycle

« Le plus court », « Courant/Dernier », « le plus long » sont les temps de cycle depuis le dernier redémarrage de la CPU.

Memory Reset de la CPU lance un effacement général

- Toutes les données (même rémanentes) sont effacées (effacement de la mémoire de travail) mémoires image processus, mémentos, compteurs, temporisations, tous les blocs de programme et de données.
- Restent conservés : le paramétrage de l'interface X1, la partie rémanente du tampon de diagnostic, l'heure.

4.4.1.2. Accès en ligne, En ligne & Diagnostic : Tampon de Diagnostic



Accès en ligne à la CPU

Si la PG et le système cible (CPU par ex.) se trouvent dans le même sous –réseau, vous avez accès à différentes fonctions « En ligne & Diagnostic » via la fonction « Abonnés accessibles ».

- Dans la fenêtre de travail de TIA Portal
- Dans la Task Card « outils en ligne »

Tampon de diagnostic

Le tampon de diagnostic est une zone de mémoire secourue de la CPU qui est organisée en anneau. Il contient tous les événements de diagnostic (messages d'erreur, alarmes de diagnostic, informations de démarrage, etc.) de la CPU dans l'ordre dans lequel ils sont apparus. La première entrée de la liste correspond au dernier événement apparu.

Tous les événements peuvent être visualisés sur la console de programmation en texte clair et dans leur ordre d'apparition.

 La taille du tampon de diagnostic est fonction de la CPU. Seule une partie du tampon de diagnostic est rémanente (l'ensemble du tampon de diagnostic n'est pas sauvegardé en cas de coupure de la tension d'alimentation).

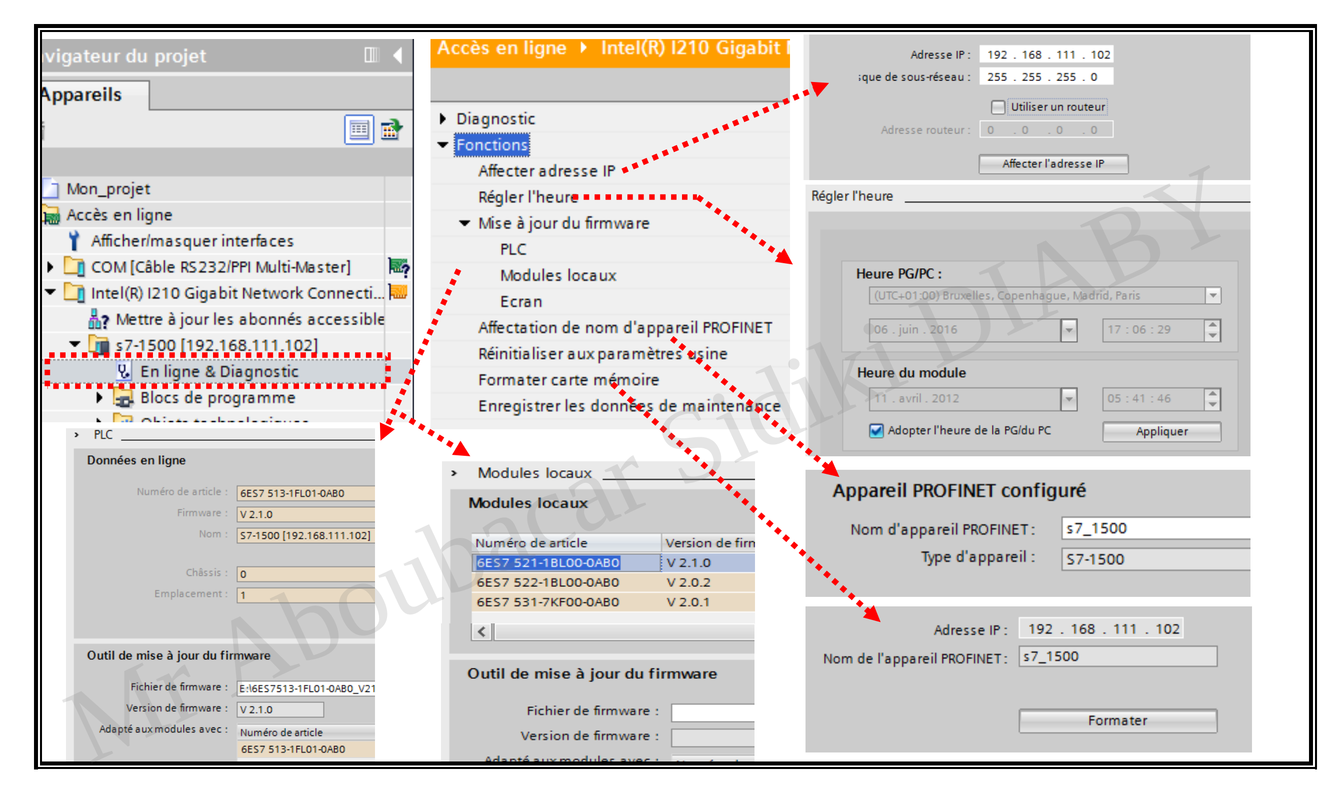
- Nombre d'entrées 1000..3200
- dont rémanentes 250...500

Détails sur l'événement

Le champ « Détails sur l'événement » fournit des informations complémentaires sur l'événement sélectionné :

- Désignation et numéro de l'événement
- Informations complémentaires en fonction de l'événement (par ex. adresse de l'instruction qui a déclenché l'événement etc.)

4.4.1.3. Accès en ligne à la CPU : adresse IP, nom, heure, mise à jour firmware, formater la carte mémoire



- **Régler l'heure**
  Chaque CPU S7 dispose d'une horloge temps réel qui peut être réglée ici.

- **Affecter l'adresse IP**
  Tant qu'une adresse IP n'a pas été définie par téléchargement d'une configuration matérielle, elle peut être affectée ou modifiée ici (cette fonction est disponible même si la PG/PC et la CPU ne sont pas affectées au même sous-réseau).

- **Réinitialiser aux paramètres d'usine**
  Contrairement à l'effacement général (MRES), cette fonction efface toutes les zones mémoire de la CPU (mémoire de travail, de chargement et rémanente, tampon de diagnostic et heure). Il est également possible d'inclure l'effacement de l'adresse IP (voir boîte de dialogue dans la figure) afin que la CPU ne dispose plus que d'une adresse MAC.

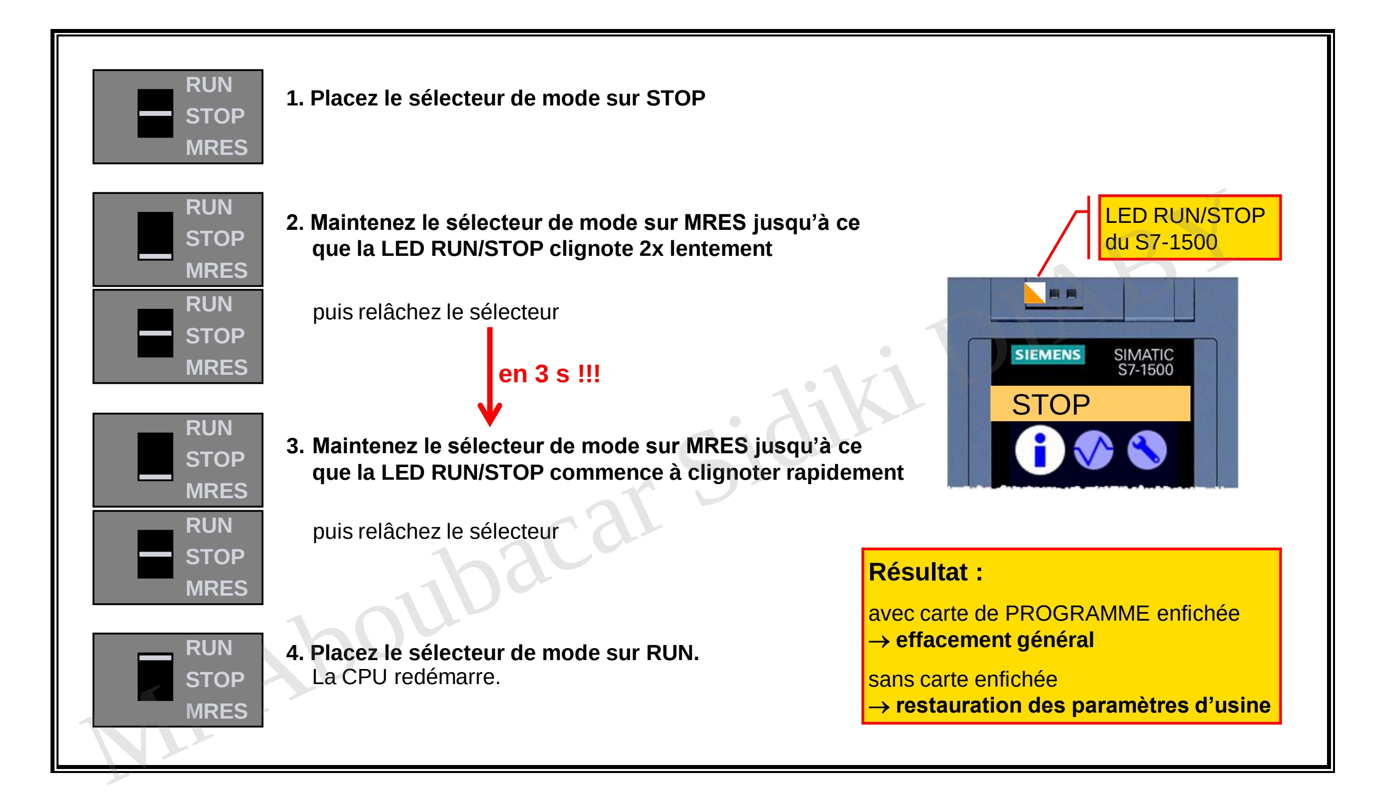
- **Formater la carte mémoire**
  La carte mémoire de la CPU peut également être effacée dans la CPU via cette fonction en ligne. La CPU ne contient alors plus que son adresse IP. Toutes les autres données (y compris la configuration des appareils) sont effacées.
  La carte ne peut pas être effacée dans le lecteur de carte via le navigateur du projet. La configuration des appareils et les blocs sont sur fond gris, ce qui signifie qu'ils sont protégés en écriture (uniquement information d'état ou ouverture par double clic).

- **Affecter un nom**
  Avec PROFINET, vous devez attribuer un nom d'appareil unique à chaque appareil qui reste enregistré de manière rémanente dans l'appareil. Ce nom d'appareil permet l'identification des modules périphériques décentralisés (PROFINET-IO) et leur remplacement sans PG/PC.

- **Mise à jour du firmware**
  Cette fonction permet d'actualiser la version du firmware de la CPU. La version courante du firmware peut être visualisée dans les données du module sous « Diagnostic -> Général ».

  
  Attention !
  Si les deux composantes doivent être mises à jour, mettez d'abord à jour l'écran et seulement ensuite la CPU.

4.5. Réinitialisation de la CPU via le sélecteur de mode



Particularité lors du Memory reset via le sélecteur de mode :

La réinitialisation de la CPU est nécessaire lors de la première mise en service d'une installation et éventuellement souhaitable lors des mises à jour du programme.

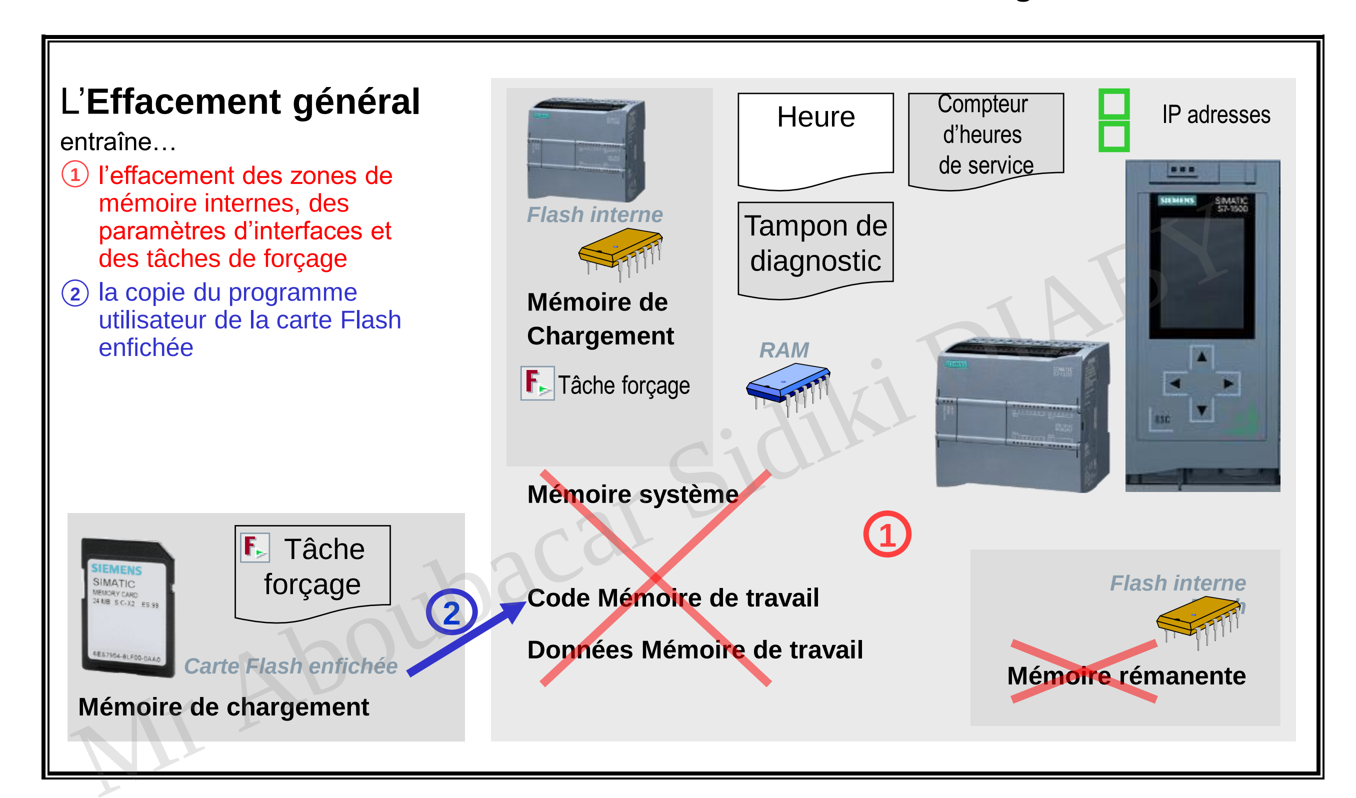


Si la CPU doit être entièrement effacée lors du montage ou d'une première mise en service, une réinitialisation aux paramètres d'usine est nécessaire. En cas d'utilisation de la mémoire de chargement sur une carte mémoire (MC), celle-ci doit être effacée dans la PG ou via un accès en ligne à la CPU avant la réinitialisation.

Lors de la réinitialisation, le système effectue les actions suivantes :

- Effacement de toutes les données utilisateur (mémoire de travail et mémoire rémanente) (mémoire image du processus, mémentos, compteurs, temporisations, ensemble des blocs de programme/données)
- Test et réinitialisation du matériel
- Vérification de la carte mémoire enfichée (MC)
  - **Carte mémoire (MC) enfichée → Effacement général**
    La CPU copie toutes les données nécessaires au traitement (configuration matérielle, blocs de programme, blocs de données) de la mémoire de chargement (de la carte mémoire) dans la mémoire de travail RAM interne. Le paramétrage de l'interface X1 et la partie rémanente du tampon de diagnostic sont conservés.
  - **Pas de carte mémoire (MC) enfichée → Réinitialisation aux paramètres d'usine**
    La CPU efface en outre : l'heure, le paramétrage de l'interface X1, le tampon de diagnostic.
    La remise en place de la MC provoque le chargement dans le mémoire de travail RAM des données nécessaires au traitement du programme issu de la mémoire de chargement : Configuration matérielle (avec adresse IP), blocs de programme, blocs de données.

4.5.1. SIMATIC S7-1200/1500 : mémoires lors de l'effacement général de la CPU



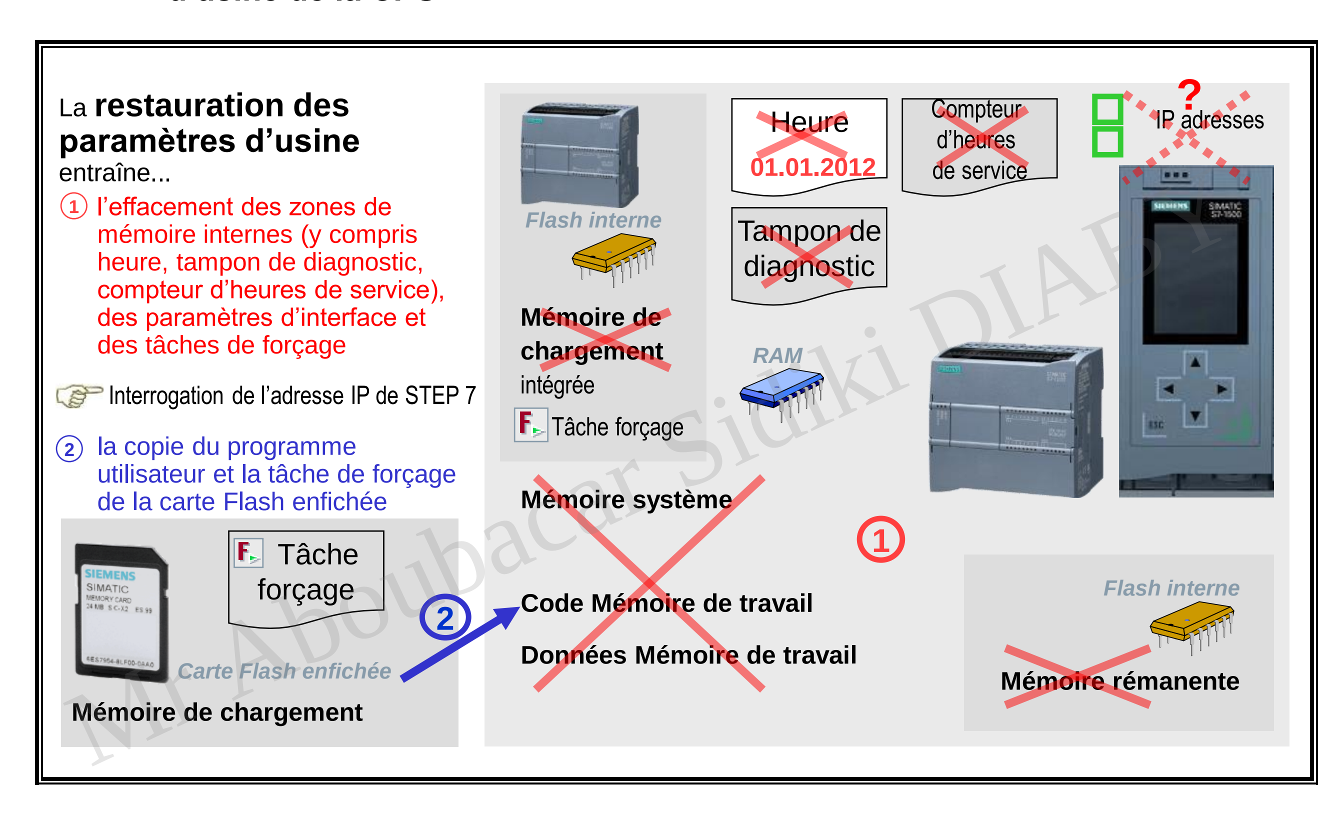
Lancement des fonctions

- Fonction en ligne STEP7 → MRES dans « Panneau de commande CPU » dans Task Card « Tester » ou « Outils en ligne »
- Afficheur (S7-1500 uniquement) → Menu principal « Paramètres », sous-menu « Réinitialiser »
- Avec le sélecteur de mode (avec carte SIMATIC enfichée)

Impact

- Interruption de la liaison en ligne entre la PG et la CPU.
- La mémoire de travail RAM est effacée, donc toutes les données utilisateurs (images du processus, bit mémoires, compteurs, temporisations, blocs de programmes et de données mêmes rémanents)
- restent: les adresses IP, le tampon de diagnostic, le compteur d'heures de fonctionnement, l'horodatage de la CPU.
- Après cela, la CPU copie toutes les données importantes pour l'exécution dans la mémoire de travail RAM à partir de la carte mémoire. (Données nécessaires à l'exécution : configuration de l'appareil, blocs de programme, blocs de données et forçages en cours).

4.5.2. SIMATIC S7-1200/1500 : mémoires lors de la réinitialisation aux paramètres d'usine de la CPU



Lancement des fonctions

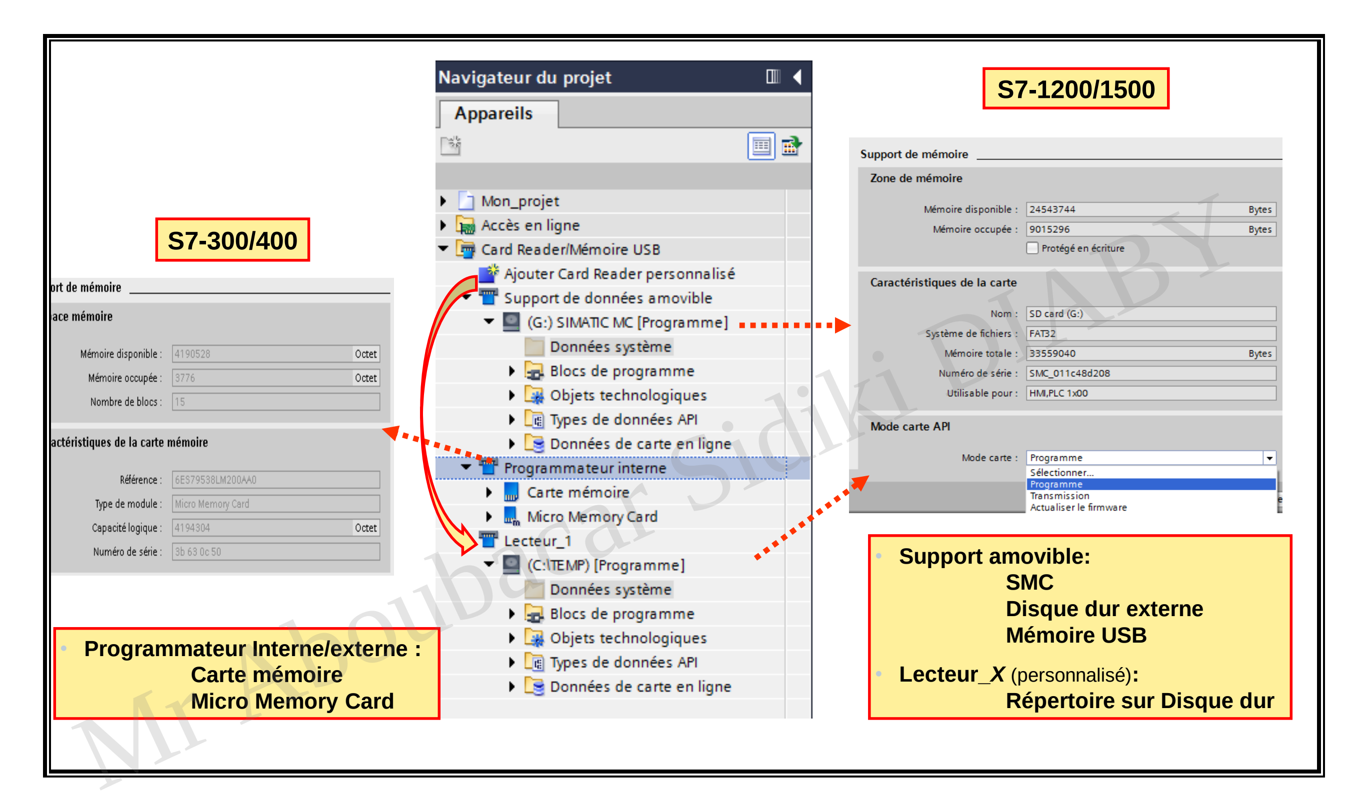
- Fonction en ligne STEP7 → MRES dans « Panneau de commande CPU » dans Task Card « Tester » ou « Outils en ligne »
- Afficheur (S7-1500 uniquement) → Menu principal « Paramètres », sous-menu « Réinitialiser »
- Avec le sélecteur de mode (uniquement sans MC)

Impact

- Interruption de la liaison en ligne entre la PG et la CPU.
- La mémoire de travail RAM est effacée, donc toutes les données utilisateurs (images du processus, bit mémoires, compteurs, temporisations, blocs de programmes et de données mêmes rémanents, tampon de diagnostic, adresses IP si sélectionnées dans STEP7.
- Les adresses IP sont conservées si cela a été retenu sous STEP7.

Avec l'insertion d'une carte mémoire (ou si elle est déjà en place), la CPU copie les données nécessaires à l'exécution dans une mémoire de travail RAM interne à partir de la carte mémoire. (Données nécessaires à l'exécution : configuration de l'appareil incluant l'adresse IP, les blocs de programmes, les blocs de données, les forçages en cours).

4.6. Lecture de la carte mémoire - SIMATIC Card Reader



Card Reader / Mémoire USB

Sous Card Reader/mémoire USB, vous accédez à la SMC insérée dans le lecteur de carte SD, au programmateur interne / externe, aux supports amovibles ou aux répertoires personnalisés .

Type de carte SIMATIC pour S7-1200/1500 :

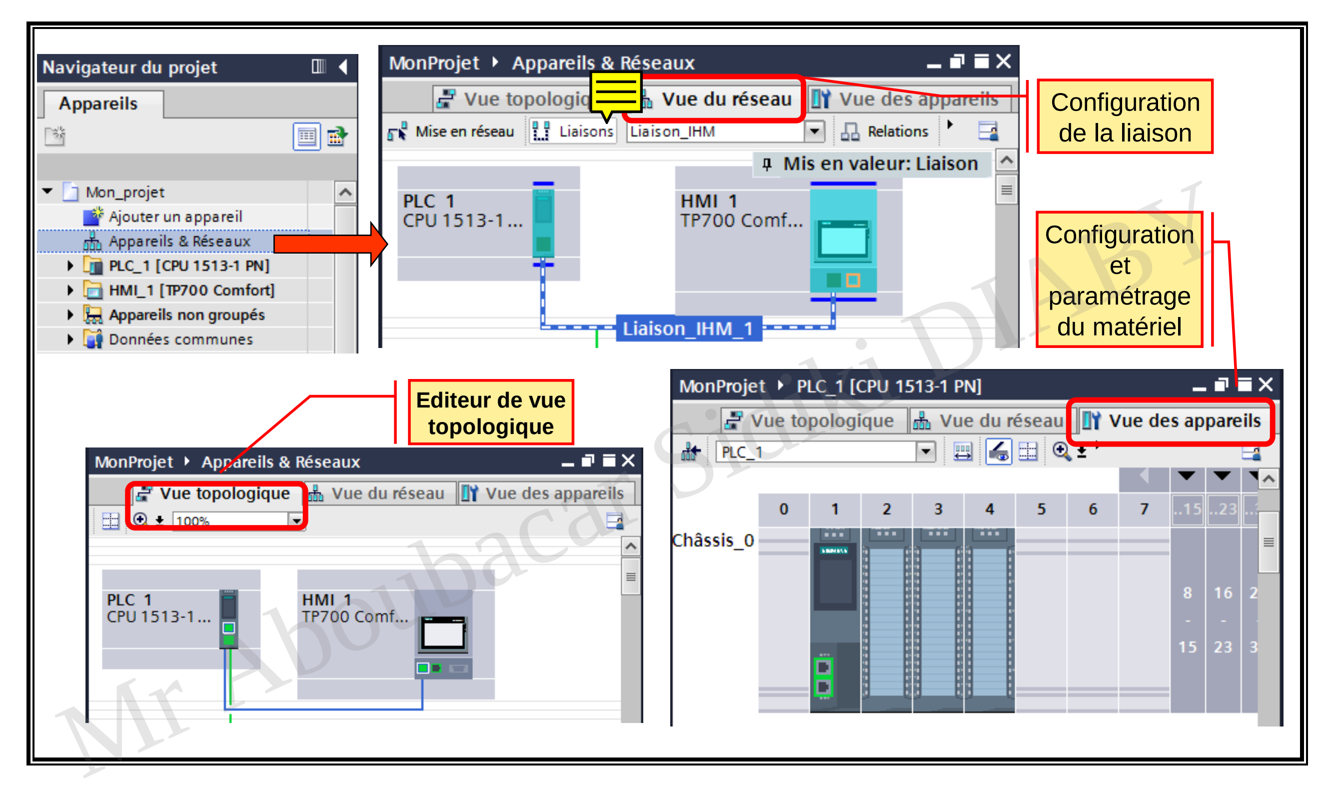
La carte SIMATIC s'utilise comme carte de programme, de transfert ou de mise à jour du firmware. Avant d'enregistrer les données voulues sur la carte SD, il convient de sélectionner le type de carte souhaité (voir figure).

- **Carte SIMATIC comme carte de programme :**
  La carte contient toutes les données de configuration et de paramétrage de la station ainsi que le programme utilisateur complet avec sa documentation. Elle doit rester enfichée dans la CPU lors du fonctionnement, car elle remplace la mémoire de chargement interne de la CPU, qui reste inutilisée.

- **Carte SIMATIC comme carte de transfert (uniquement pour le S7-1200) :**
  La carte contient les mêmes données que la carte de programme, mais ne doit pas rester enfichée lors du fonctionnement. Après enfichage de la carte et mise sous tension, toutes les données de la carte sont copiées dans la mémoire de chargement interne de la CPU. La carte doit ensuite être retirée et le système redémarré.

- **Carte SIMATIC pour la mise à jour du firmware :**
  La carte SIMATIC contient les fichiers nécessaires à la mise à jour du firmware. Une fois l'opération accomplie (voir instructions du fichier texte joint), la carte SIMATIC doit être retirée.

4.7. Composantes de l'éditeur « Appareils & Réseaux »



L'éditeur « Appareils & Réseaux » est constitué d'une vue des appareils, d'une vue du réseau et d'une vue topologique.

Vue du réseau

La vue du réseau sert à mettre en réseau les appareils.

- Configuration des liaisons

Vue des appareils

La vue des appareils sert à configurer les appareils.

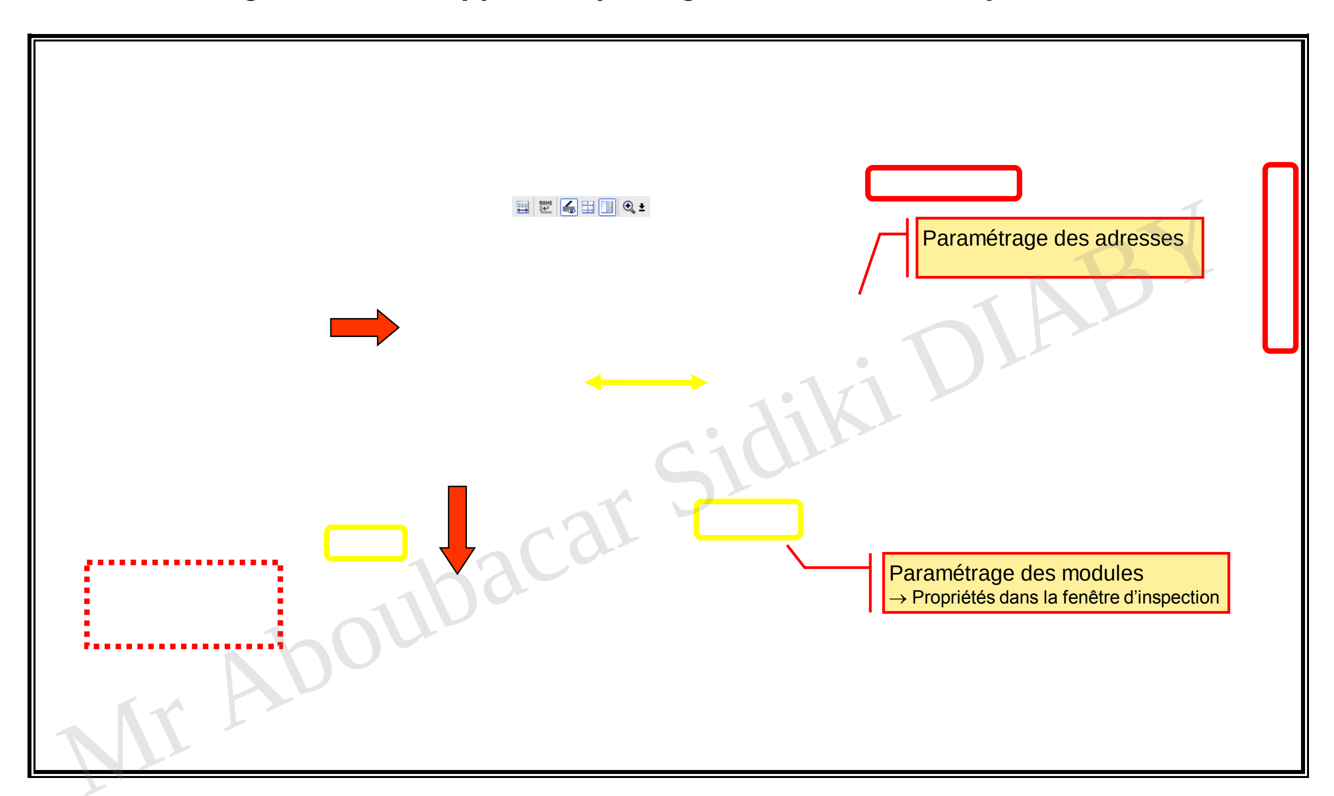
- Configuration matérielle
- Paramétrage des modules

Vue topologique

La vue topologique sert à définir la structure physique des réseaux.

- Configuration des interfaces affectées et des relations entre appareils voisins
- Comparaison en ligne- hors ligne ainsi que la synchronisation des affectations de ports et de voisinage
- La topologie permet l'échange d'un appareil sans affectation de nom

4.7.1. Configuration des appareils (configuration du matériel)



Créer et éditer la configuration du matériel

- « Matériel et réseaux » éditeur → sélection module matériel → sélection onglet « Matériel »
- Ouvrir l'éditeur « Configuration matériel » depuis l'arborescence projet.

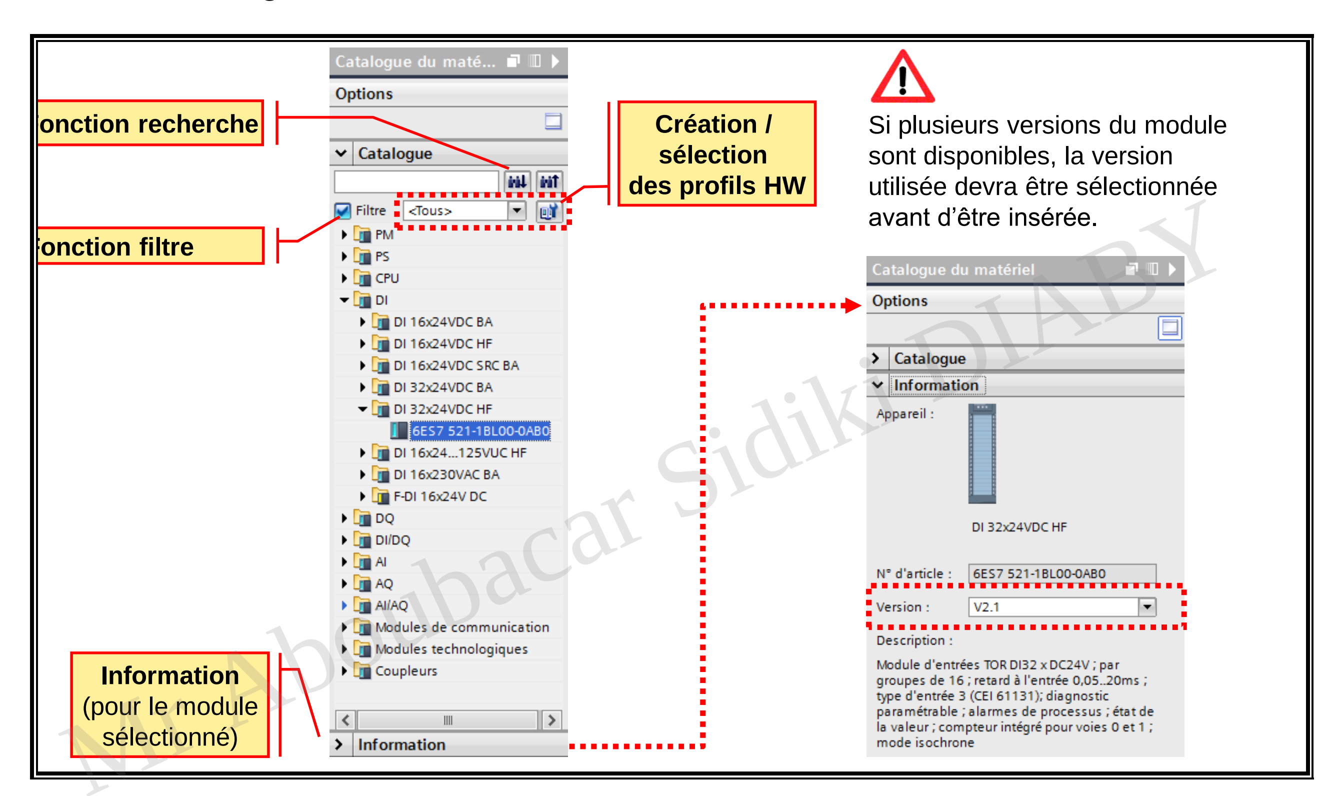
Composants de l'éditeur « configuration matériel »

- **Onglet « Vue matériel » dans la fenêtre de travail**
  Cette éditeur est fait de deux parties. La section droite peut-être ouverte et fermée par ce bouton ou le volet de séparation des deux sections peut-être positionné avec la souris à la largeur désirée.
– Fenêtre gauche = configuration du module
– Fenêtre droite = réglages des adresse et paramètres des modules configurés.
- **Onglet « Propriétés » dans la fenêtre d'inspection**
  Cet onglet est utilisé pour régler les paramètres des modules sélectionnés dans la fenêtre de travail. Ici toutes les propriétés ou les paramètres des modules sélectionnés sont affichés et peuvent être modifiés. Dans la partie gauche de cette vue, la section de navigation est organisée en groupes de paramètres pour faciliter leur accès.
- **« Catalogue matériel, vue « task card »**
  Contient le catalogue matériel pour la configuration sélectionné dans la fenêtre de travail.

Arborescence projet → « modules locaux »

Dans l'arborescence projet, les modules et leurs paramètres, composant une configuration, sont accessibles dans le dossier « modules locaux ».

4.7.2. Catalogue du matériel



Le Catalogue du matériel contient tous les appareils et composants matériels sous forme d'arborescence. Il est possible de faire glisser les appareils ou modules souhaités de ce catalogue vers la fenêtre de travail graphique de l'éditeur « Appareils & Réseaux ».

Fonction de recherche

Cette fonction facilite la recherche des composants matériels souhaités. La recherche peut également porter sur les textes de description des modules.

Fonction filtre

 activé

Seuls s'affichent les modules adaptés au contexte courant (fenêtre de travail).

 désactivé

Tous les objets présents dans le catalogue s'affichent.

Contenu du Catalogue du matériel

- Vue du réseau → uniquement les objets pouvant être mis en réseau
- Vue des appareils → tous les modules appartenant à l'appareil courant de la fenêtre de travail

Lors du passage de la vue du réseau à la vue des appareils, la vue des objets filtrés est adaptée à l'appareil sélectionné dans la vue du réseau.

Profil

Il est possible de créer un profil du catalogue, ce qui permet d'étendre la fonction de filtrage.

Information

La palette « Information » affiche des informations de détail sur l'objet sélectionné dans le catalogue.

- Représentation schématique
- Nom
- Référence
- Numéro de version
- Brève description du module

4.8. Insérer une station matérielle

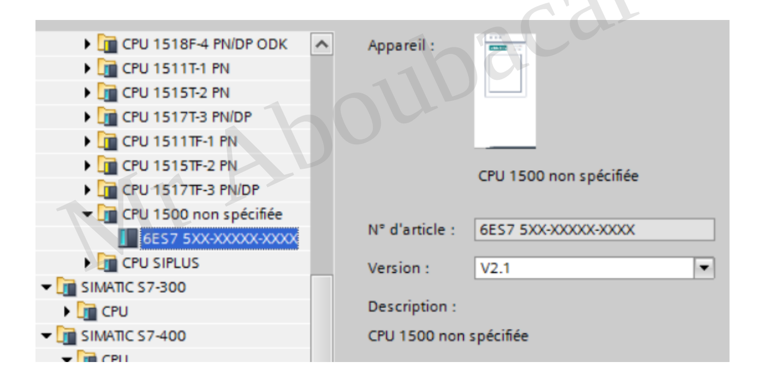


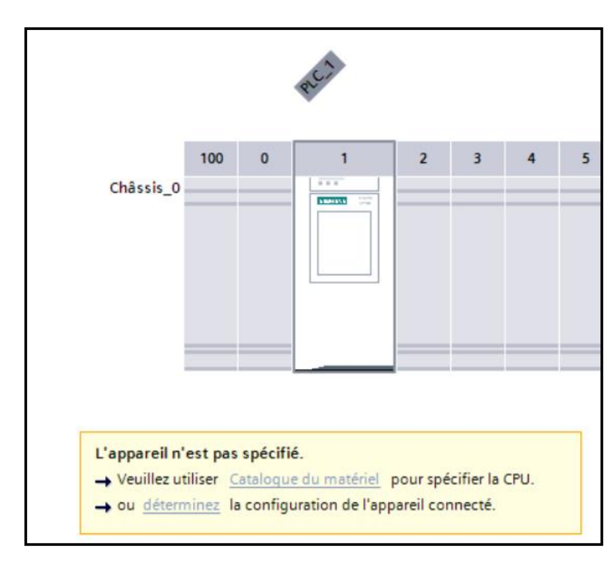
Avec la configuration des appareils, l'utilisateur détermine la disposition des modules dans le châssis (rack).
Lors de la création d'un nouvel appareil, un châssis adapté est automatiquement créé. L'appareil sélectionné est enfiché au premier emplacement autorisé du châssis.

Indépendamment du chemin choisi, l'appareil ajouté est visible dans la vue des appareils et la vue du réseau de l'éditeur « Appareils & Réseaux ».

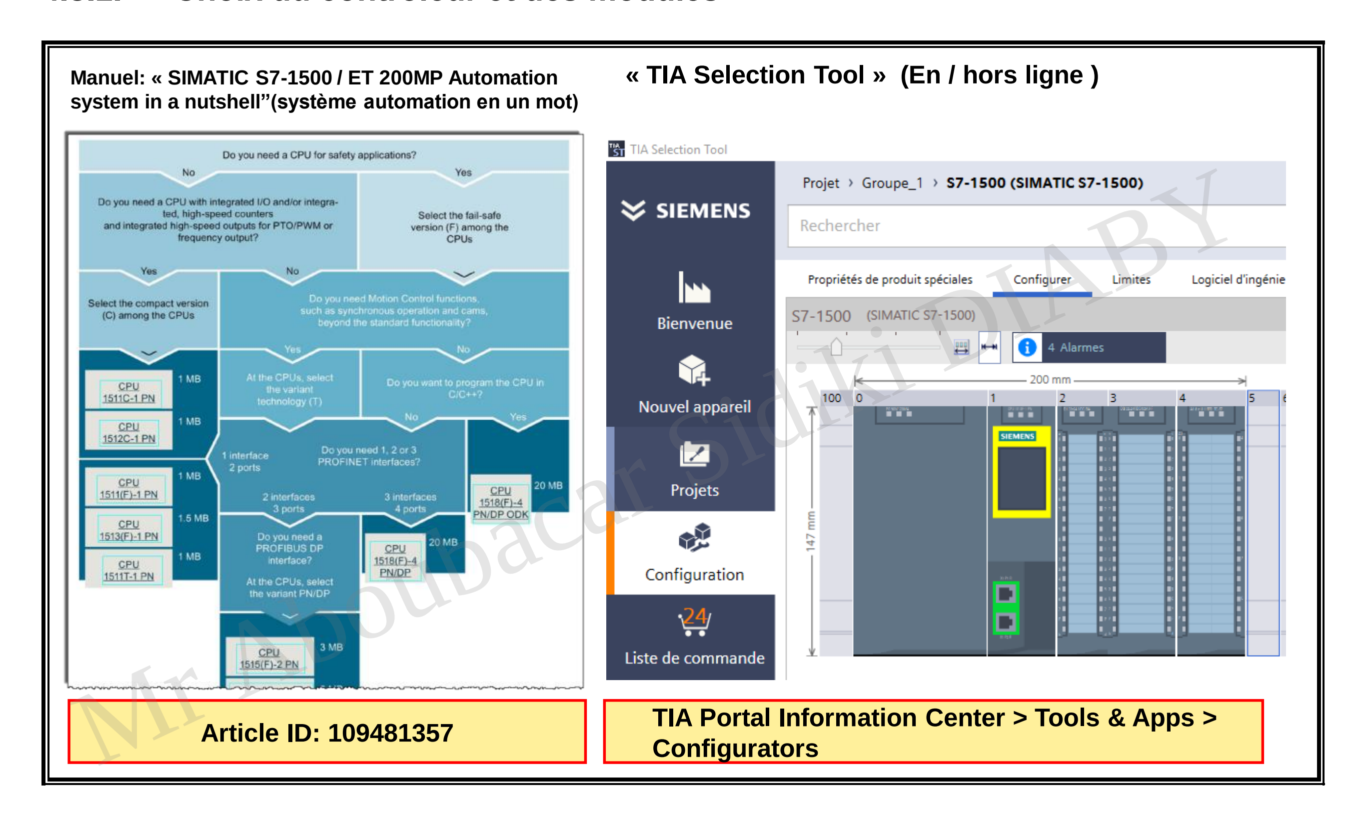
CPU non spécifiée

Pour le choix du contrôleur, vous pouvez également retenir une « CPU 1500 non spécifiée ». Cela est nécessaire lorsque la version de la CPU n'est pas encore fixée mais que vous aimeriez démarrer la programmation. Ici, vous allez juste spécifier le firmware de la CPU qui sera utilisé par la suite. Le matériel utilisé pourra être configuré ultérieurement à partir du catalogue du matériel ou déterminé en ligne.





4.8.1. Choix du contrôleur et des modules

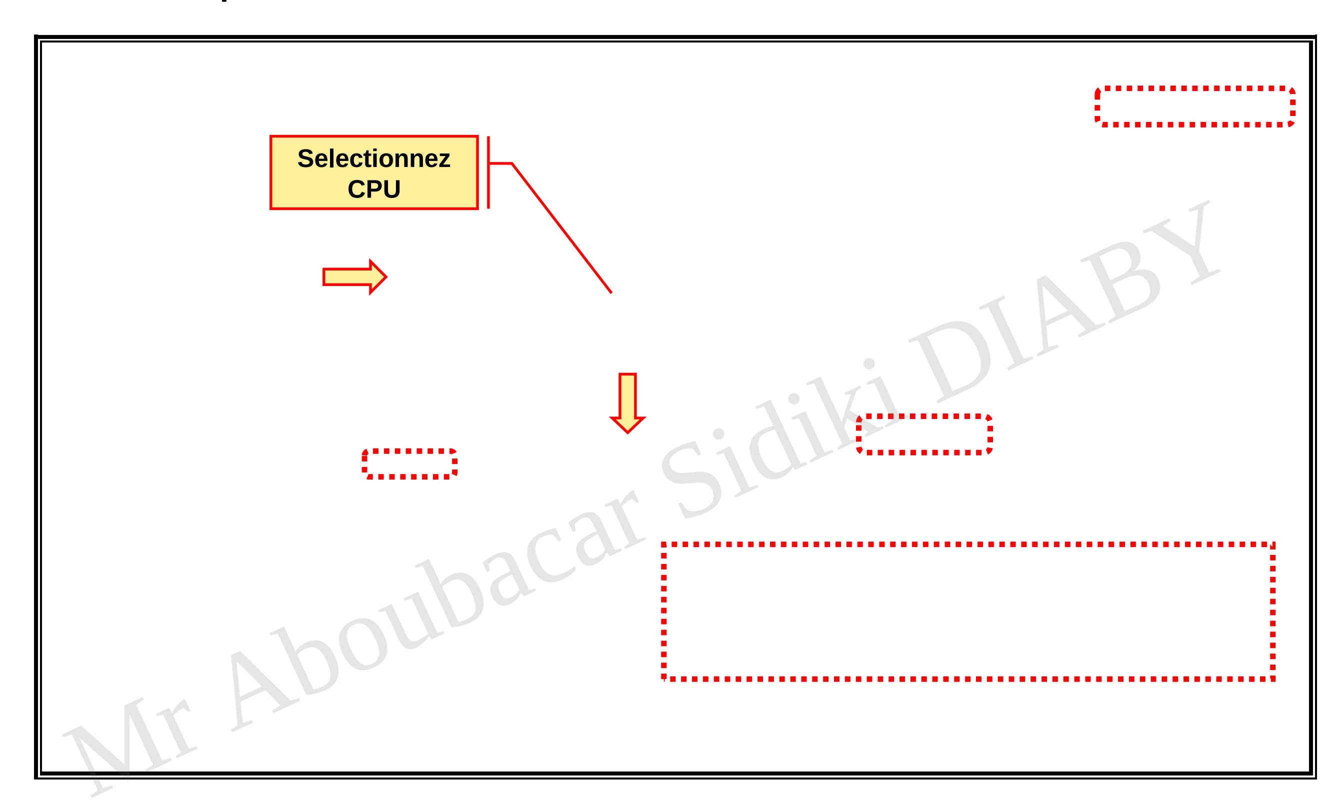


SIMATIC S7-1500 vous permet d'intégrer une large gamme de CPUs. Vous pouvez agencer chaque CPU avec des modules d'E/S, des modules de communication et des modules technologiques. Si, par exemple, la taille mémoire et les performances d'une CPU1511-1 PN sont suffisantes pour vous, alors vous la compléter avec des modules de communication pour PROFINET et PROFIBUS. Pour les fonctions technologiques vous disposez des CPUs technologiques, de modules technologiques en complément aux CPUs compactes.

Pour choisir le contrôleur adéquat vous pouvez utilisez le manuel « **SIMATIC S7-1500 / ET 200MP Automation system In a nutshell » (SIMATIC S7-1500 / ET 200MP sytème d'automatisation en un mot)** qui contient des guides de choix complémentaires. Il est disponible sur notre site sous le numéro d'article ID: 109481357.

Vous disposez également de l'outil « **TIA Selection Tool** » qui vous guide dans votre sélection, configuration et commande des produits du Totally Integrated Automation. Après la configuration du matériel dans «TIA Selection Tool » vous disposez d'une liste regroupant l'ensemble des composants matériels nécessaires (modules, connecteurs, câbles, rails profilés etc.)
De plus, une commande peut être directement lancée via Industry Mall à partir du « TIA Selection Tool ».

4.8.2. Propriétés de la CPU : adresse Ethernet



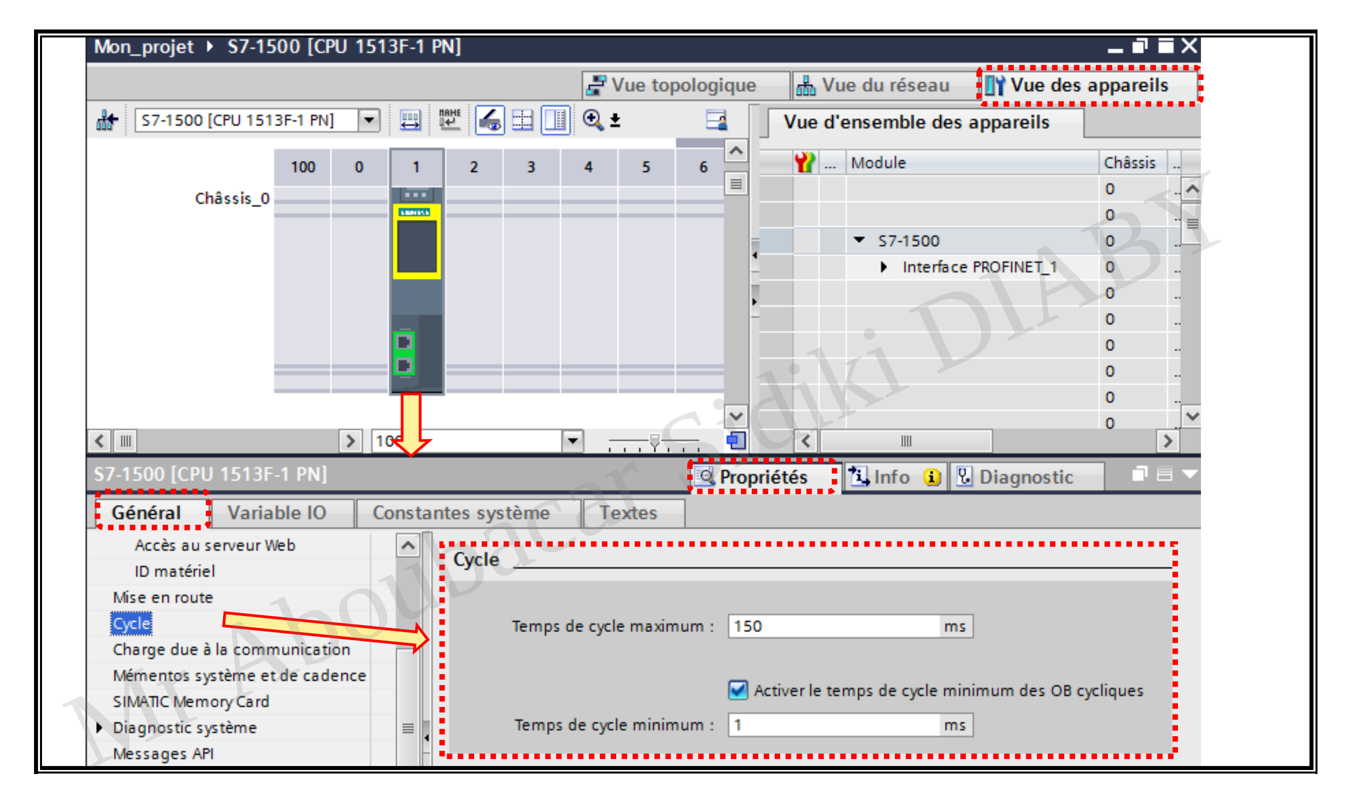
Interface PROFINET

Que l'éditeur « Appareils & Réseaux » se trouve dans la vue des appareils ou dans la vue du réseau, il est possible de paramétrer l'interface CPU-PROFINET sous l'onglet « Propriétés » après avoir sélectionné la CPU dans la fenêtre d'inspection.



Si une liaison en ligne entre console de programmation et CPU doit être établie, il faut affecter aux deux appareils le même masque de sous-réseau et des adresses IP situées dans le même sous-réseau.

4.8.2.1. Propriété de la CPU : temps de cycle maximum



Temps de cycle

Le temps de cycle est le temps nécessaire à la CPU pour traiter une fois l'ensemble du programme cyclique. Ce temps de cycle n'est pas le même à chaque cycle, car des parties du programme utilisateur peuvent également être traitées de manière conditionnelle et le traitement du programme peut en outre être interrompu (par ex. par une alarme de diagnostic, une alarme d'horloge, une alarme de processus etc.)

Temps de cycle maximum

Le système d'exploitation surveille le temps de traitement du programme en fonction du temps limite maximal paramétré. Lorsque le temps de traitement du programme dépasse le temps de surveillance du cycle paramétré

- le système d'exploitation appelle l'OB d'erreur correspondant (OB80)
- le système d'exploitation inscrit l'évènement dans le tampon de diagnostic
- le système d'exploitation active la LED ERROR de la CPU

S7-1200

La CPU reste à l'état RUN même si aucun OB80 n'est programmé et chargé.
Lorsque le temps de traitement du programme dépasse le double du temps de surveillance de cycle paramétré (Erreur 2xMaxCycleTime), la CPU passe à l'état STOP sans appeler l'OB80.

S7-1500

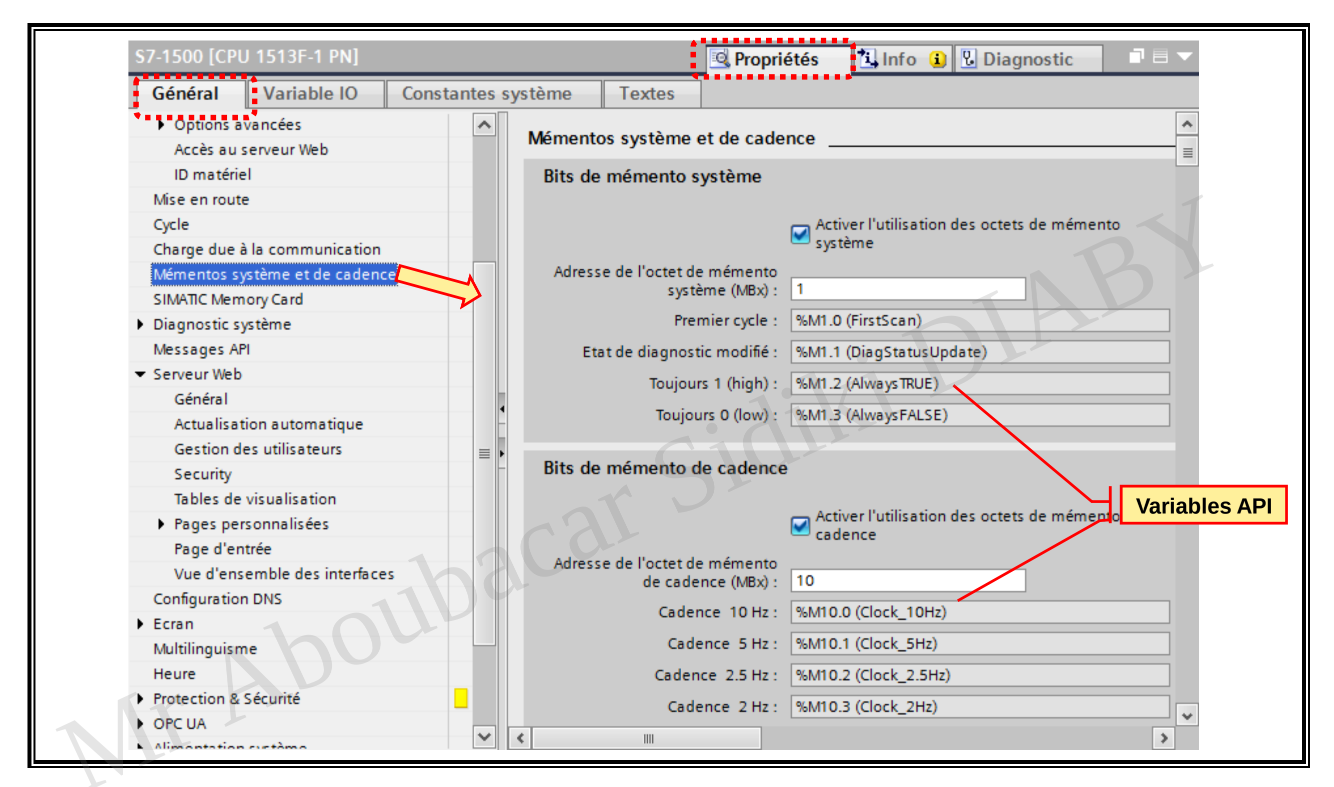
Si un OB80 est programmé et chargé, la CPU reste à l'état RUN, sinon elle passe à l'état STOP

L'instruction RE_TRIGR permet de réactiver la surveillance du temps de cycle ou de la remettre à zéro

Temps de cycle minimum

Le temps de cycle minimum est le temps minimal qui doit s’écouler pour traiter une fois le programme utilisateur cyclique et l’actualisation de la périphérie correspondante. Le lancement du cycle de CPU suivant est retardé si ce temps n’est pas encore écoulé.

4.8.2.2. Propriété de la CPU : mémentos système et de cadence



Une variable API est automatiquement créée pour chaque bit de mémento système ou de cadence disponible.

Mémentos de cadence (8 bits)

Il s'agit de bits de mémento dont l'état binaire est modifié périodiquement par le système d'exploitation de la CPU dans un rapport impulsion-pause de 1:1. La figure montre les différentes fréquences. Les mémentos de cadence s'utilisent pour déclencher des opérations périodiques (par ex. faire clignoter un voyant lumineux).



Attention !

Les mémentos de cadence fonctionnent de manière asynchrone par rapport au cycle de la CPU. Avec des temps de cycle longs, l'état du mémento de cadence peut se modifier plusieurs fois au cours d'un même cycle.

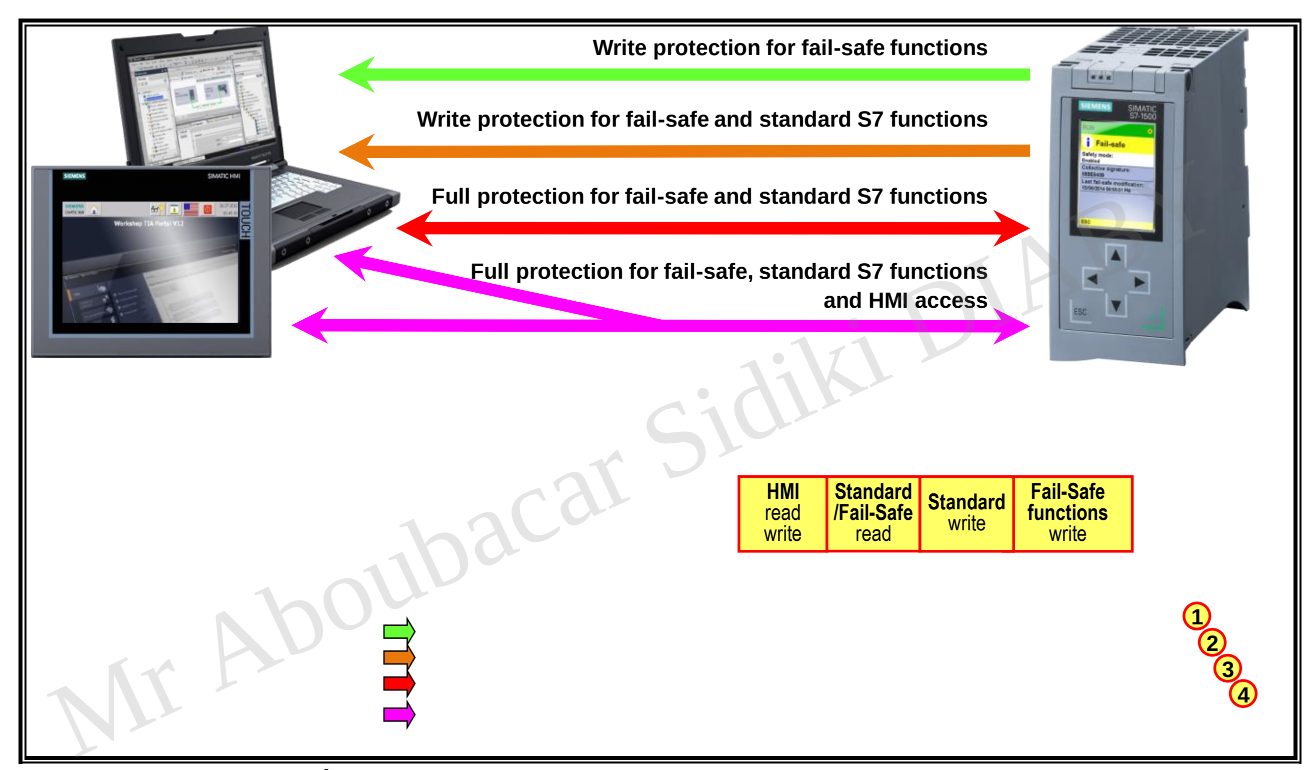
Mémentos système (4 bits)

Il s'agit de bits de mémento qui fournissent une information d'état du système.

- « FirstScan » =1 lors du premier cycle de la CPU ; sinon =0
- « DiagStatusUpdate » =1 en présence d'un problème ; sinon =0
- Un mémento 0 statique et un mémento 1 statique

Les 4 autres bits sont réservés pour d'éventuelles extensions ultérieures.

4.8.2.3. Propriétés de la CPU : protection





Attention !
Le mot de passe est stocké de manière cryptée dans la CPU et ne peut pas être supprimé.



Les fonctions protégées par mot de passe ne peuvent être exécutées que par une seule PG/PC à un instant donné. Aucune autre PG/PC ne peut alors s'identifier.

Les niveaux de protection

Les niveaux de protection suivants définissent les droits d'accès (lecture/écriture) pour la console de programmation à la CPU :

- Accès complet incluant la sécurité (pas de protection): → sélection par défaut pour les CPU-F
  L'accès en lecture et en écriture est toujours autorisé.
- Accès complet (pas de protection): → sélection par défaut pour les CPU non F
  L'accès en lecture est toujours autorisé, l'accès en écriture est uniquement autorisé pour le programme standard.
- Accès en lecture: → protection en écriture
  L'accès est uniquement possible en lecture. Aucune donnée ne peut être modifiée dans la CPU, ni chargement de blocs, ni configuration matérielle modifiée, ni affectation de paramètre ne peuvent être transférée vers la CPU sans saisir un mot de passe.
- Accès HMI: → protection en écriture et lecture pour STEP7
  Aucun accès en écriture et lecture n'est autorisé à partir de l'ingénierie. Seul le type de CPU et les données d'identification peuvent être affichées dans le navigateur du projet sous « Abonnés accessibles ». L'affichage d'informations en ligne ou de blocs sous « Abonnés accessibles » n'est pas possible sans la saisie d'un mot de passe.
- Aucun accès (protection complète) : → protection générale en écriture et lecture pour STEP7 et HMI
  L'accès en écriture et lecture est également verrouillé pour les appareils HMI sans un mot de passe configuré dans la liaison.

Autorisation d'accès par mots de passe

Dans l'exemple présenté, « Aucun accès (protection complète) » est sélectionné. Cela signifie que sans un mot de passe ni STEP7 ni les appareils HMI ne peuvent accéder soit en lecture soit en écriture à la CPU.

Les niveaux de protection ci-dessus peuvent néanmoins être annulés par la saisie de mots de passe:

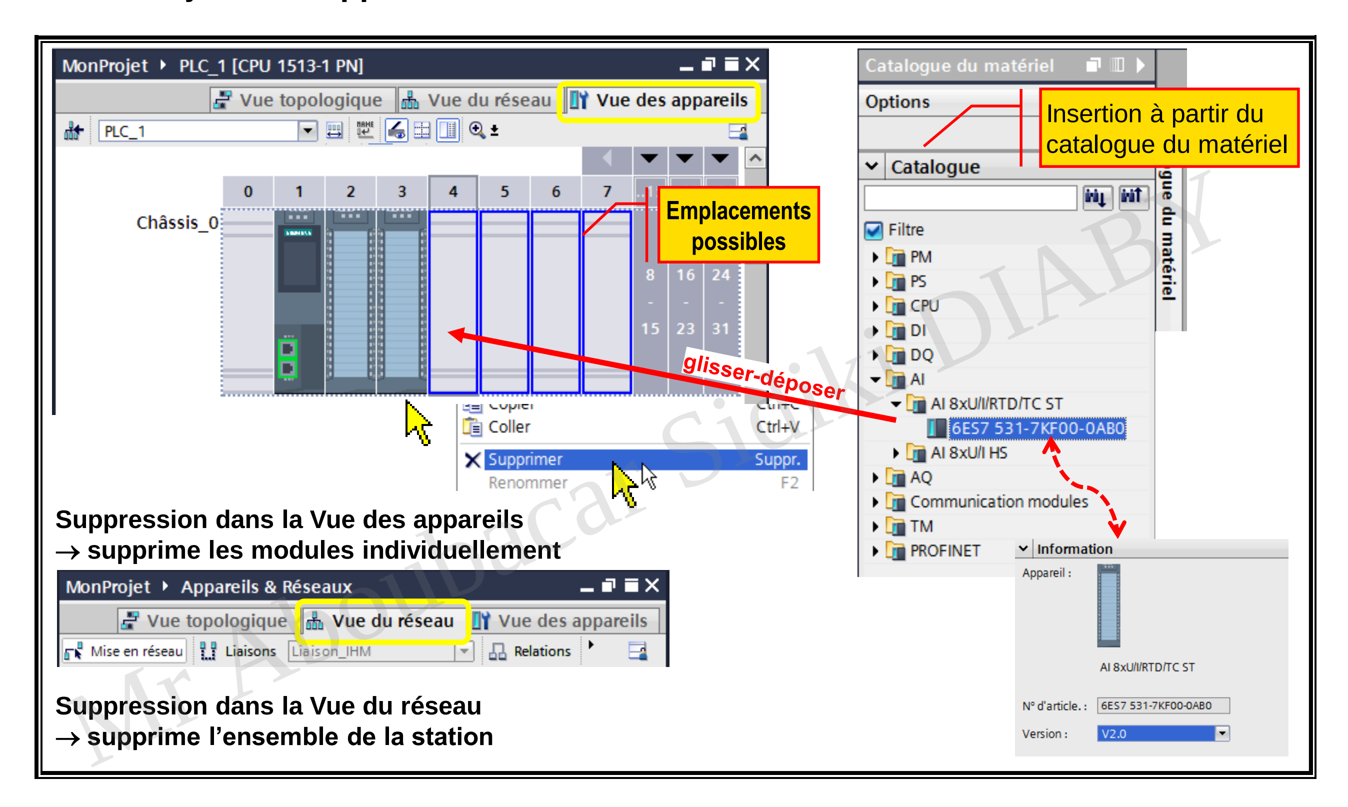
- En spécifiant un mot de passe (4) un appareil HMI retrouve son accès en lecture et écriture pour la CPU. Par contre STEP7 n'a pas d'accès en lecture ou en écriture.
- En spécifiant un mot de passe (3) un appareil HMI retrouve son accès en lecture et écriture pour la CPU, pour STEP7 seul les accès en lecture sont autorisés, pas d'accès en écriture.
- En spécifiant un mot de passe (2) les accès en lecture et en écriture pour le programme standard de la CPU sont autorisés aussi bien pour l'appareil HMI que pour STEP7.
- En spécifiant un mot de passe (1) les accès en lecture et en écriture sont autorisés pour les programmes pour l'appareil HMI et STEP7.

Autorisation d'accès via la communication PUT/GET :

- Pour autoriser l'accès à la CPU pour d'autres contrôleurs avec les fonctions PUT et GET il faut autoriser l'accès dans les propriétés de la CPU sous « Protection et Sécurité » > « Mécanismes de connexion ».



4.8.3. Ajouter / supprimer un module



Insérer

Un nouveau module peut être enfiché par glisser-déposer même entre des modules déjà enfichés. Il suffit pour cela de faire glisser le nouveau module par le haut entre les deux modules déjà enfichés. Une marque d'insertion apparaît et le nouveau module peut être inséré. Tous les modules déjà enfichés sont alors décalés d'un emplacement vers la droite. Les modules éventuellement surnuméraires sont déposés dans la zone de stockage des modules non enfichés.

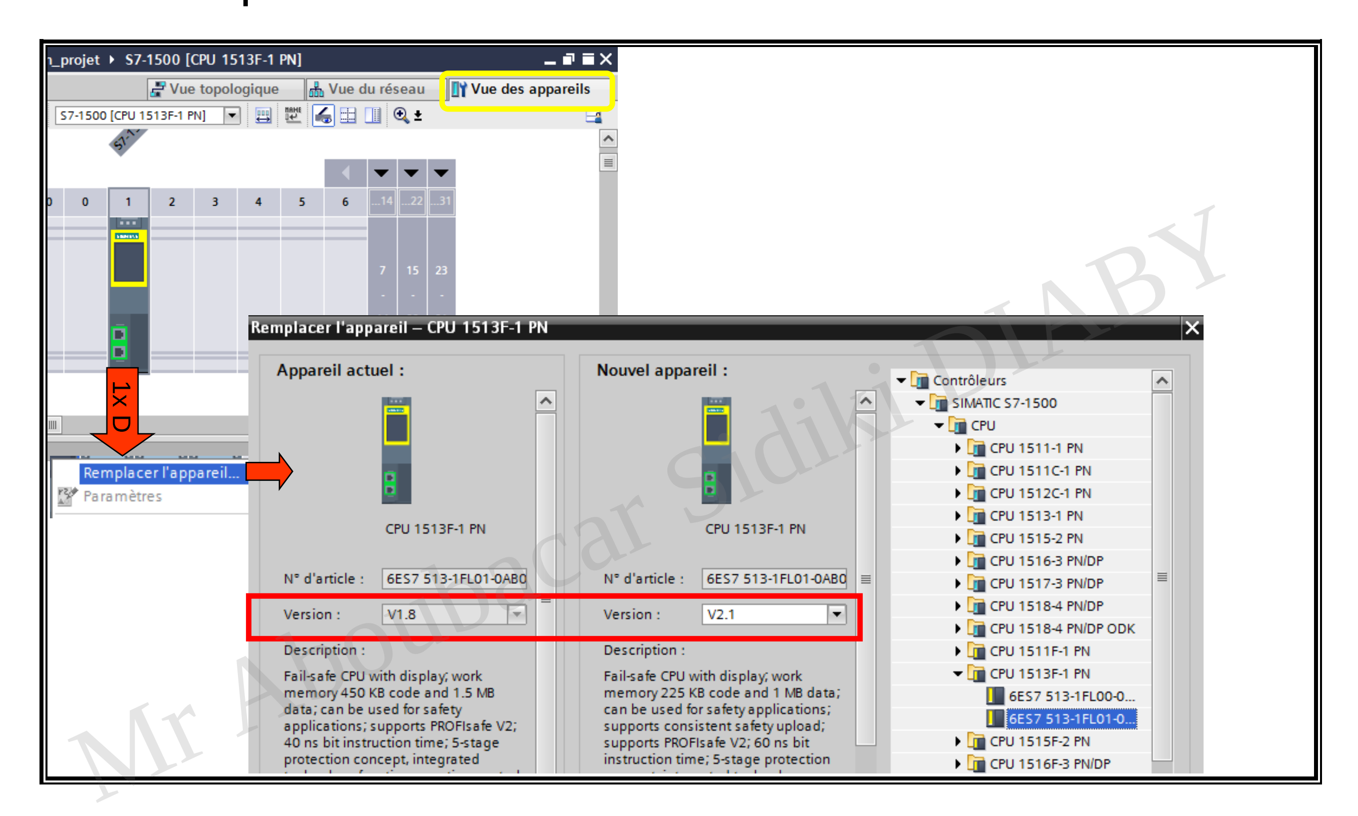
Supprimer

Les composants matériels supprimés sont effacés du système et les adresses occupées sont libérées.

Choisir la version

Lors du choix du module, il faut tenir compte de sa version spécifiée dans la partie information de la « Task Card » du catalogue matériel.

4.8.4. Remplacer un module

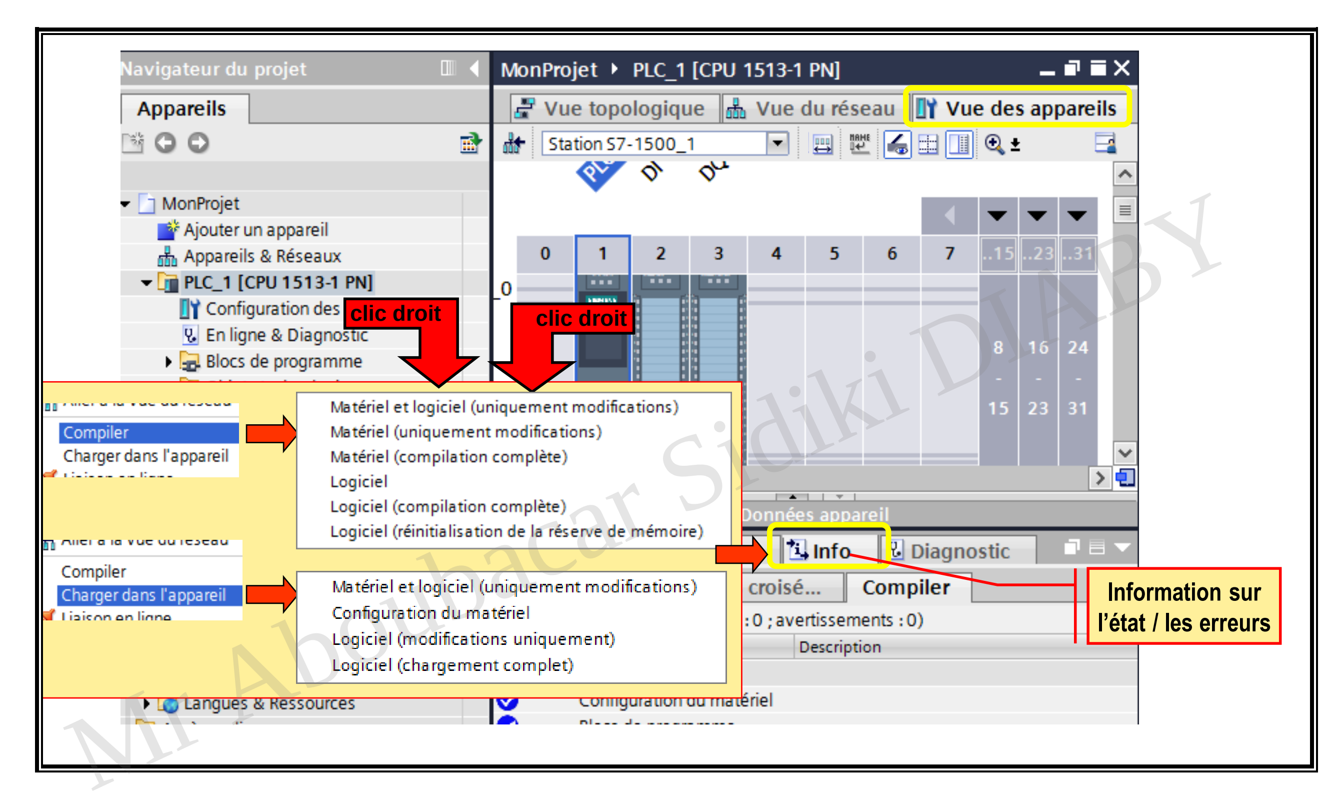


Remplacer

L'avantage d'un remplacement de module par rapport à une suppression suivie d'une insertion de nouveau module réside dans le fait que l'ensemble des paramètres de l'ancien module sont dans ce cas récupérés par le nouveau module. Un remplacement de module peut par ex. être nécessaire après une mise à jour du firmware pour adapter la version de la CPU dans le projet hors ligne à la version de la CPU (en ligne).
Un remplacement n'est possible qu'entre composants compatibles.

Il est également possible de remplacer un module par « glisser-déposer » du module récent à partir du catalogue du matériel vers l'ancien module.

4.8.5. Compiler et charger la configuration matérielle



Les composants suivants d'une station matérielle peuvent être compilés ou chargés :

- Matérielle et logicielle (uniquement modifications)

  L'ensemble des modifications de la configuration matérielle, paramétrage et du programme sont compilés/chargés

- Matériel (uniquement modifications)
  Seul l'ensemble des modifications de la configuration matérielle est compilé/chargé.

- Matériel (compilation complète)
  Seul l'ensemble de la configuration matérielle et du paramétrage est compilé/chargé.

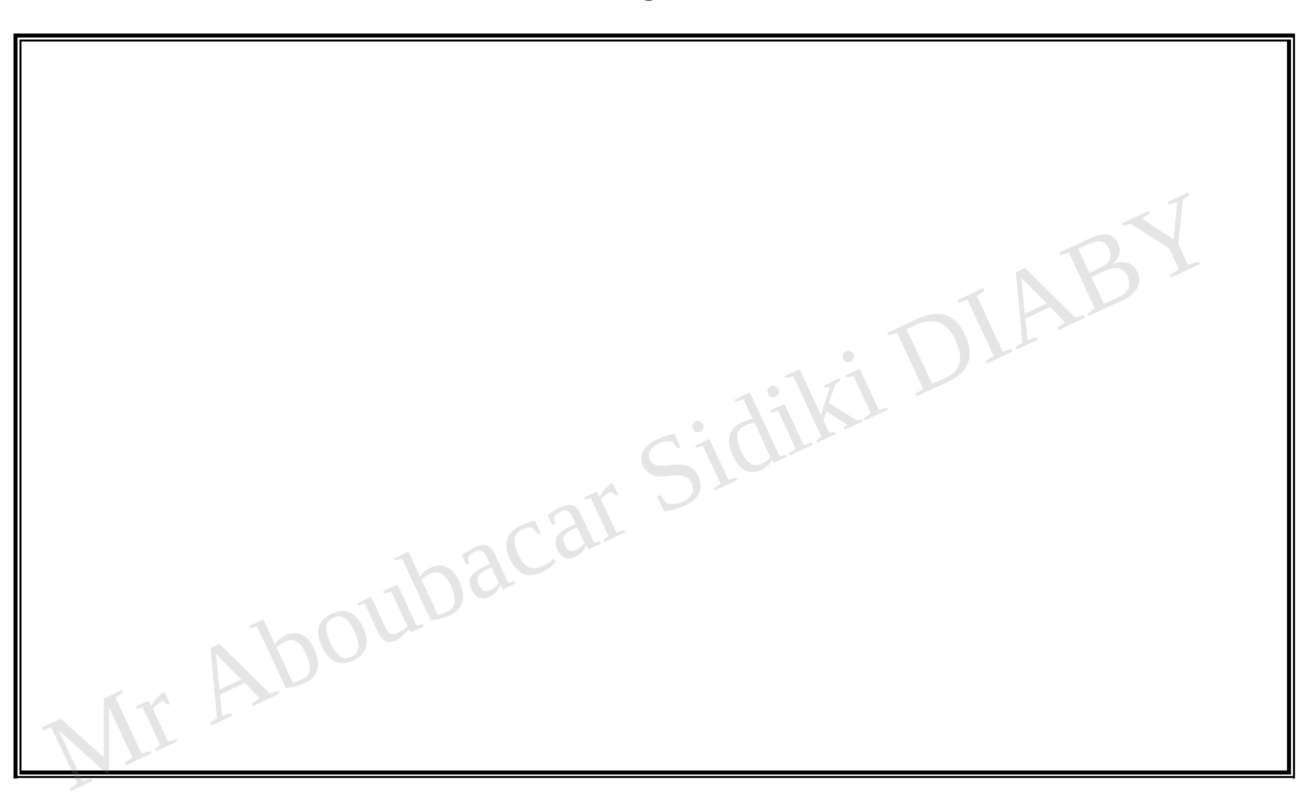
- Logiciel (uniquement modifications)
  Seuls les blocs modifiés du programme utilisateur sont compilés/chargés.

- Logiciel (compilation complète)
  Tous les blocs du programme utilisateur sont compilés/chargés.

  Note: le choix « compilation complète »est réservé pour des développements futurs et réalise pour l'instant la fonction identique à la compilation complète des blocs.

- Logiciel (réinitialisation de la réserve de mémoire)
  Tous les blocs modifiés du programme utilisateur sont compilés, les blocs de données qui utilisent de la réserve de mémoire (à cause de l'insertion de nouvelles variables) sont réinitialisés. Les données courantes des blocs de données modifiés sont écrasées par les valeurs initiales lors du prochain chargement et les réserves de mémoire sont libérées même si une réserve de mémoire (pour le chargement d'une modification de structure de DB avec maintien de la valeur) est activée dans les propriétés du bloc de données.

4.9. Exercices : Création et chargement de la station S7-1500



Enoncé

L'objectif est de créer et charger une station de type S7-1500 correspondant à votre banc pédagogique avec le paramétrage correct pour les exercices à venir

Mr Aboubacar Sidiki DIABY

4.9.1. Exercice 1 : raccorder la PG à la CPU et paramétrer l'adresse IP de la PG

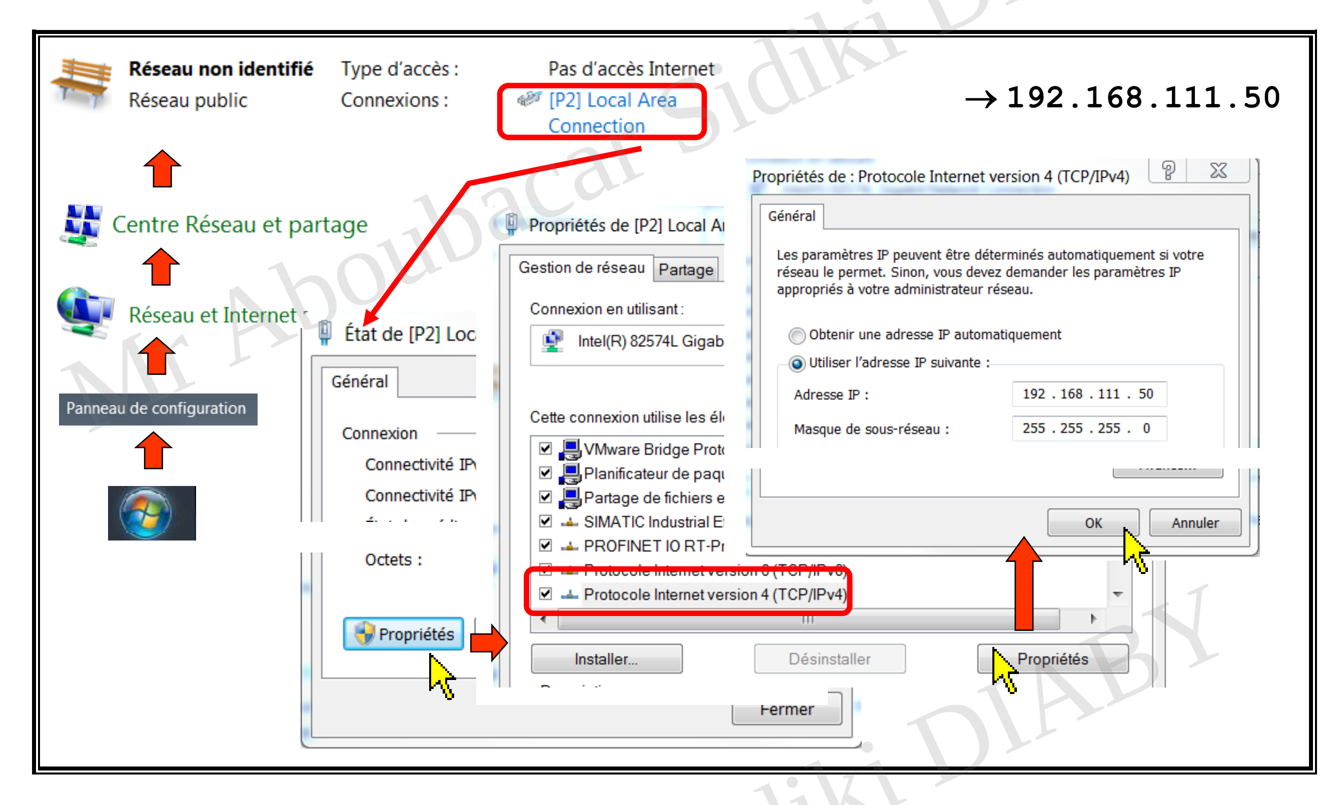
Pour relier la PG à la CPU, vous devez utiliser l'interface de gauche (vue de l'avant) de la PG (P2=LocalAreaConnection).

Reliez cette interface à la connexion « P1 » de l'appareil de formation via un câble Ethernet.

Actualiser la vue d'ensemble des modules

→ Liaison PG et adresse IP de la PG
→ Vue d'ensemble des modules du chapitre 1 – Exercice 2

Reportez la liaison Ethernet et l'adresse IP de la liaison PG dans la vue d'ensemble.



Enoncé : Affectez à la PG pour cette interface l'adresse IP 192.168.111.50 et le masque de sous-réseau 255.255.255.0.

Marche à suivre

1. Appeler l'éditeur Windows « Centre Réseau et partage »

>> Démarrer > Panneau de configuration > Réseau et Internet > Centre Réseau et partage

2. Contrôler/établir le câblage PG ↔ Valise de formation

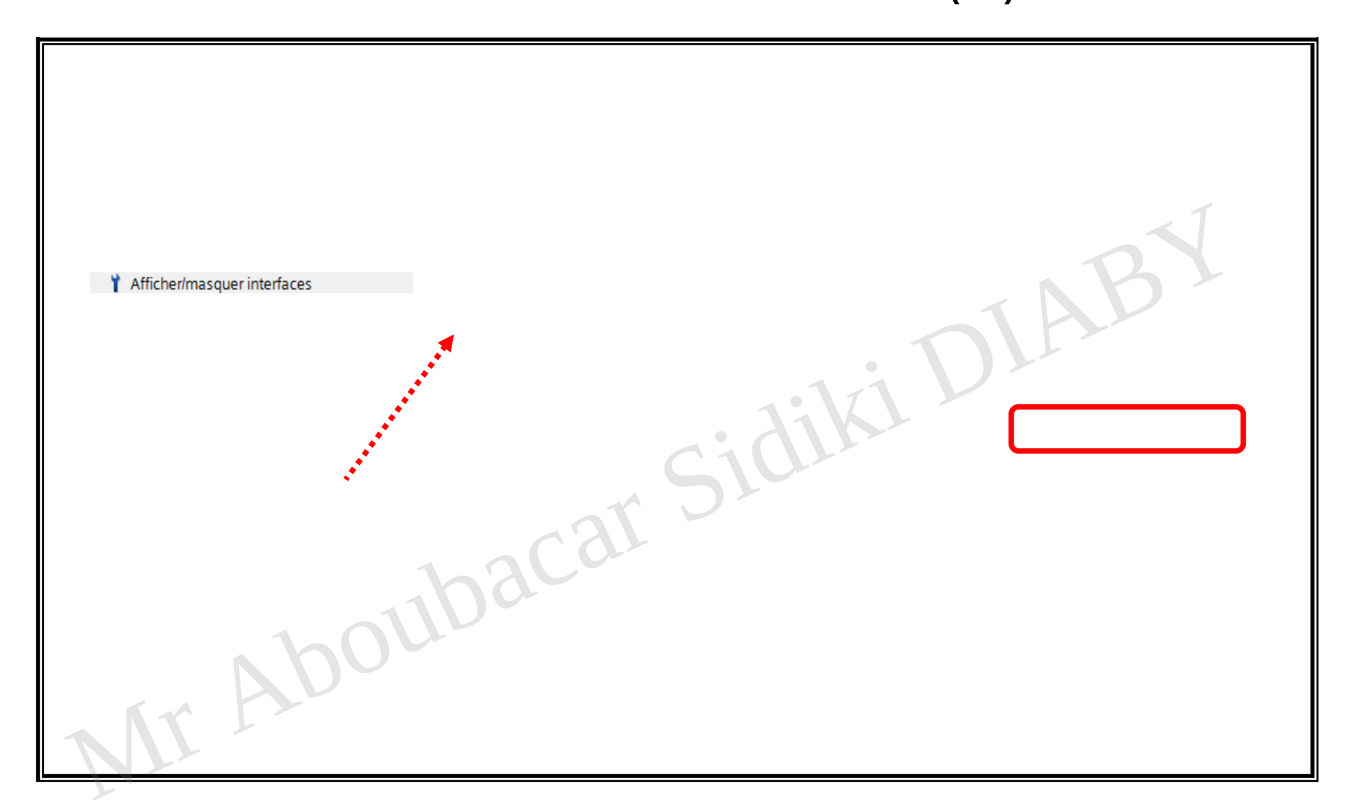
La liaison n'est directement visible dans « Centre Réseau et partage » que si une liaison avec la CPU est établie.

Reliez l'interface Ethernet de gauche de la PG avec le connecteur P2 de l'appareil de formation. L'appareil de formation est encore sous tension à la suite du précédent exercice.

3. Modifier les propriétés de l'interface

Affectez l'adresse requise à la « [P2]Local Area Connection » pour le « Protocole Internet version 4 (TCP/IPv4) ».

4.9.2. Exercice 2 : Effacer le contenu de la carte mémoire (SD) de la CPU



Enoncé

Pour effacer complètement la CPU, vous devez effacer la carte mémoire SIMATIC de la CPU. Cette carte mémoire peut être effacée de 3 manières différentes :

- Via l'explorateur Windows (en insérant la carte dans le lecteur SD de la console de programmation) ⚠ PAS DE FORMATAGE DE LA CARTE
- Via le TIA Portal (en insérant la carte dans le lecteur SD de la console de programmation)
- Via le TIA Portal (en laissant la carte mémoire dans la CPU)

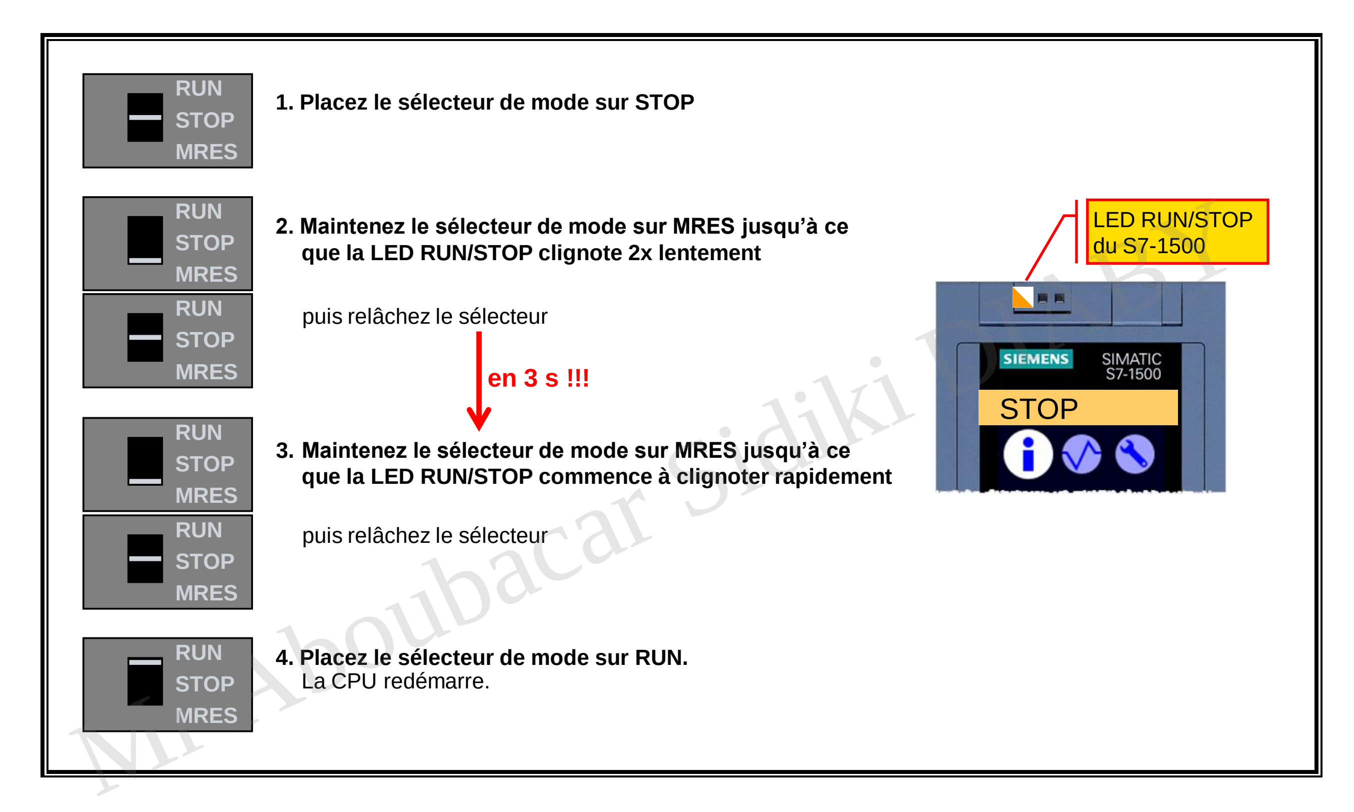
Marche à suivre

1. Vous vérifiez que la carte SD soit bien insérée dans la CPU.
2. Vous recherchez les abonnées accessibles à partir de votre interface de communication (Intel® 82574…..).
3. Sous S7-1500, cliquez sur En ligne & Diagnostic *(Voir diapositive ci-dessus)*
4. Sous Fonctions → Formater carte mémoire cliquez sur le bouton Formater pout lancer le formatage de la carte

Remarque :

Si la CPU est protégée par un mot de passe que vous ne connaissez pas, vous devez formater la carte en l'insérant directement sur la console de programmation. (coté gauche)

4.9.3. Exercice 3 : Effacement général de la CPU



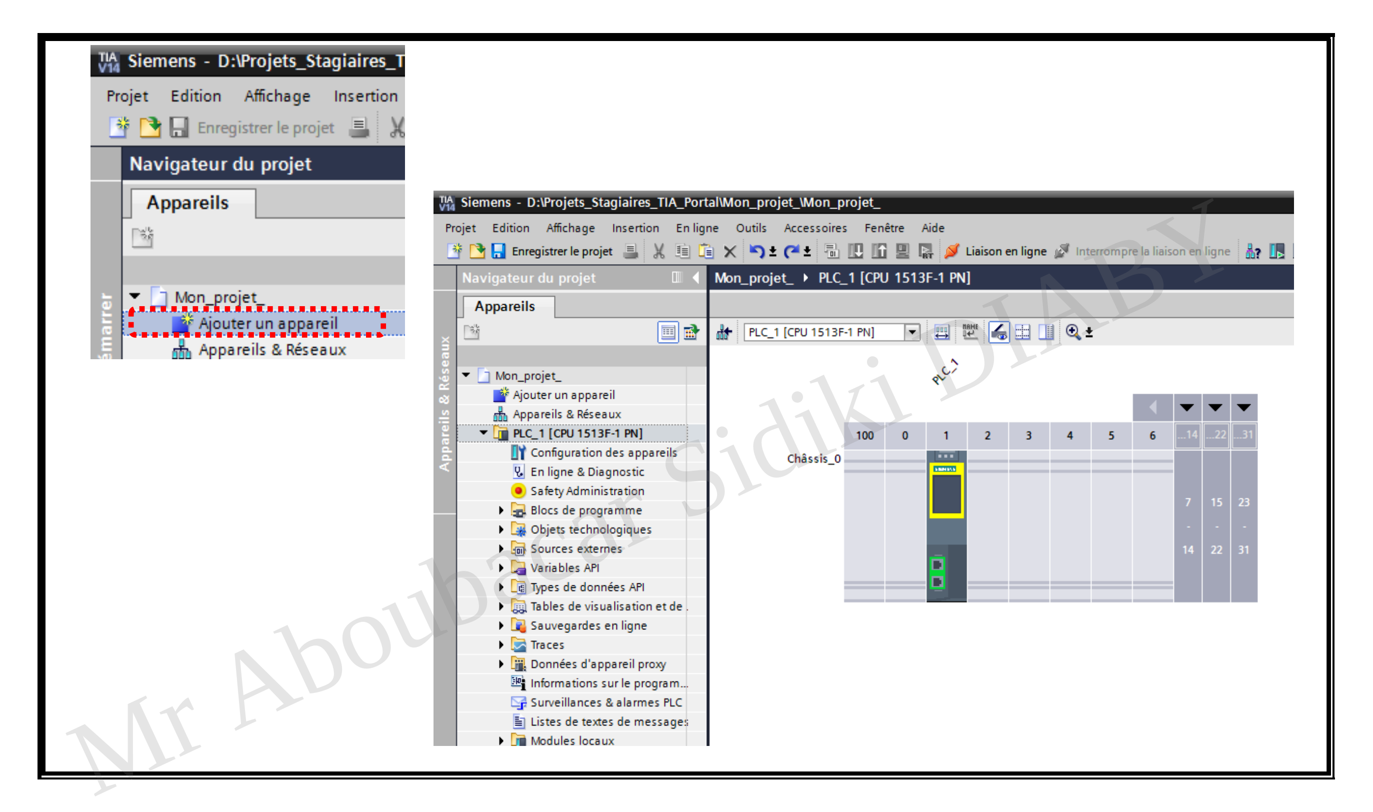
Enoncé

Après avoir effacé la SD de la CPU dans le cadre du précédent exercice, vous devez à présent effacer aussi la mémoire de travail de la CPU. Vous devez pour ce faire procéder à un effacement général.

Marche à suivre

1. Effectuez l'effacement général directement sur la CPU en suivant les étapes indiquées sur la figure.
2. Redémarrez la CPU en faisant passer le sélecteur de mode de fonctionnement de STOP à RUN.

4.9.4. Exercice 4: Création d'un nouveau projet et ajout d'un appareil (Contrôleur)



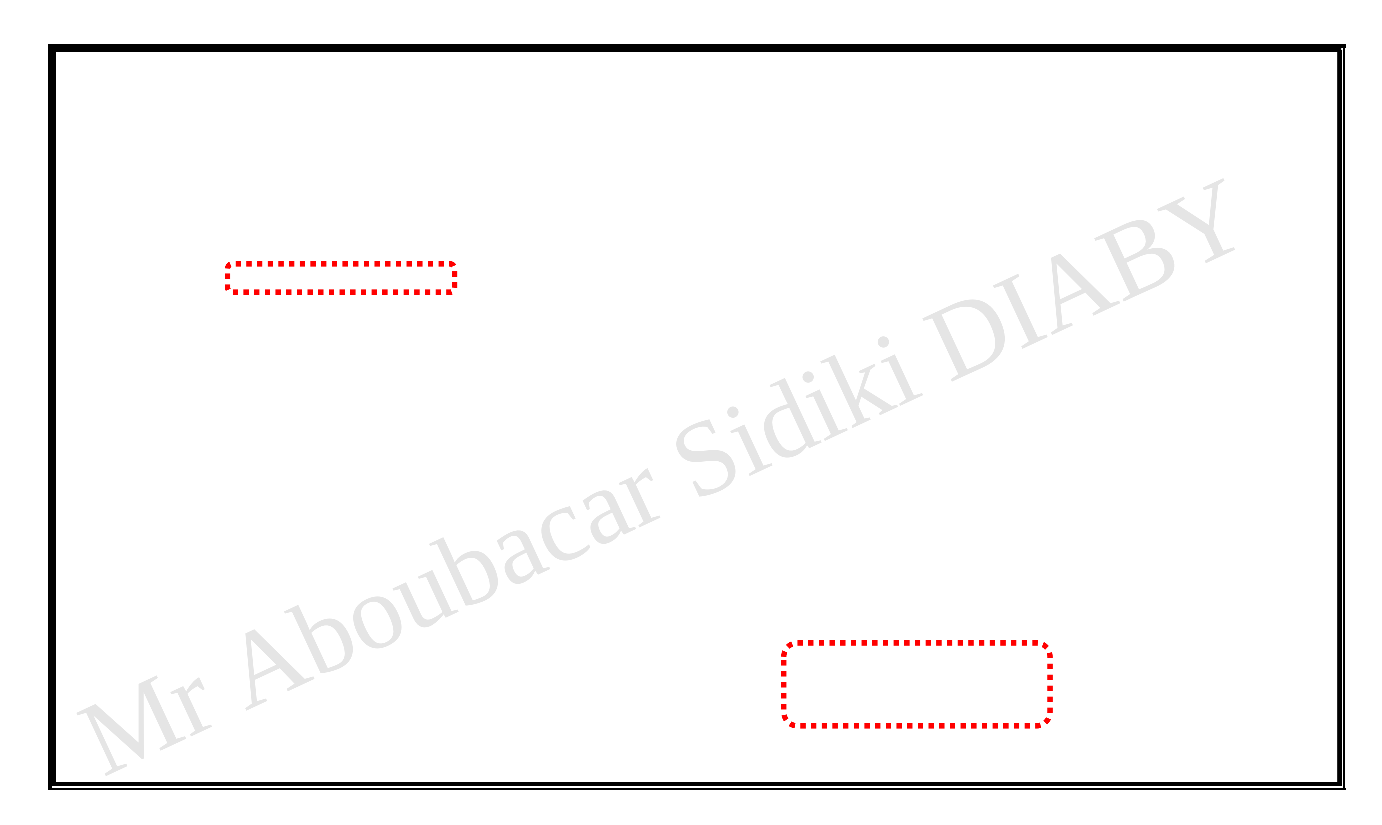
Enoncé

Vous allez créer un nouveau projet ayant pour nom « Monprojet2 » et ajouter un appareil

Marche à suivre

1. Enregistrez votre projet.
2. Créez un nouveau projet « Mon_projet ».
3. Ajoutez le contrôleur correspondant à votre poste de travail.
   (Dans l'exercice 1 du chapitre 2 « Vue d'ensemble » vous aviez déjà récupéré et noté les informations nécessaires à partir de l'afficheur.)

4.9.5. Exercice 5: renommer le contrôleur et désactiver « F-activation »



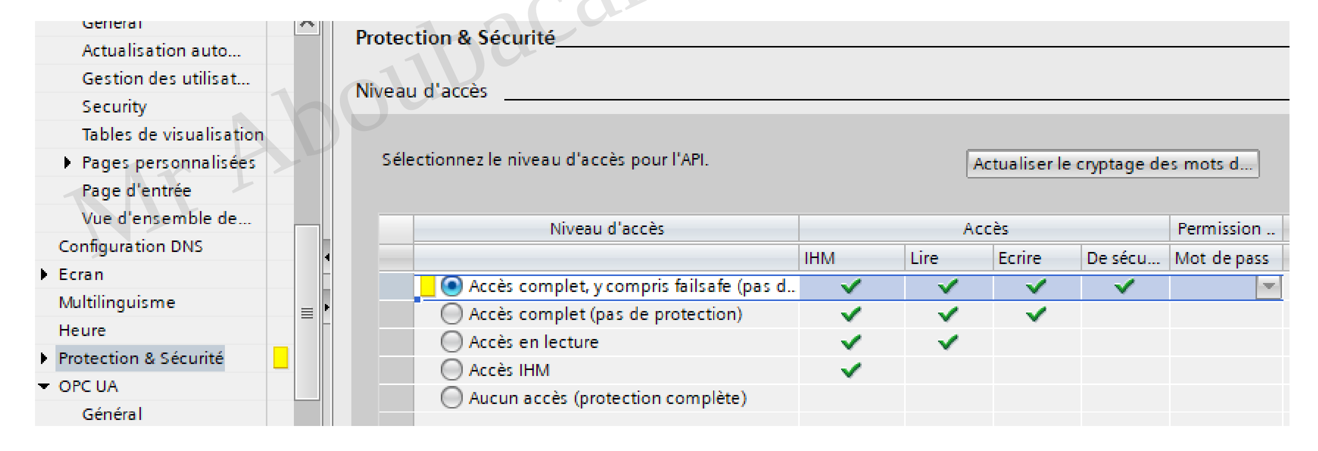
Enoncé

Changez le nom du contrôleur et désactivez « F-activation ».

Marche à suivre

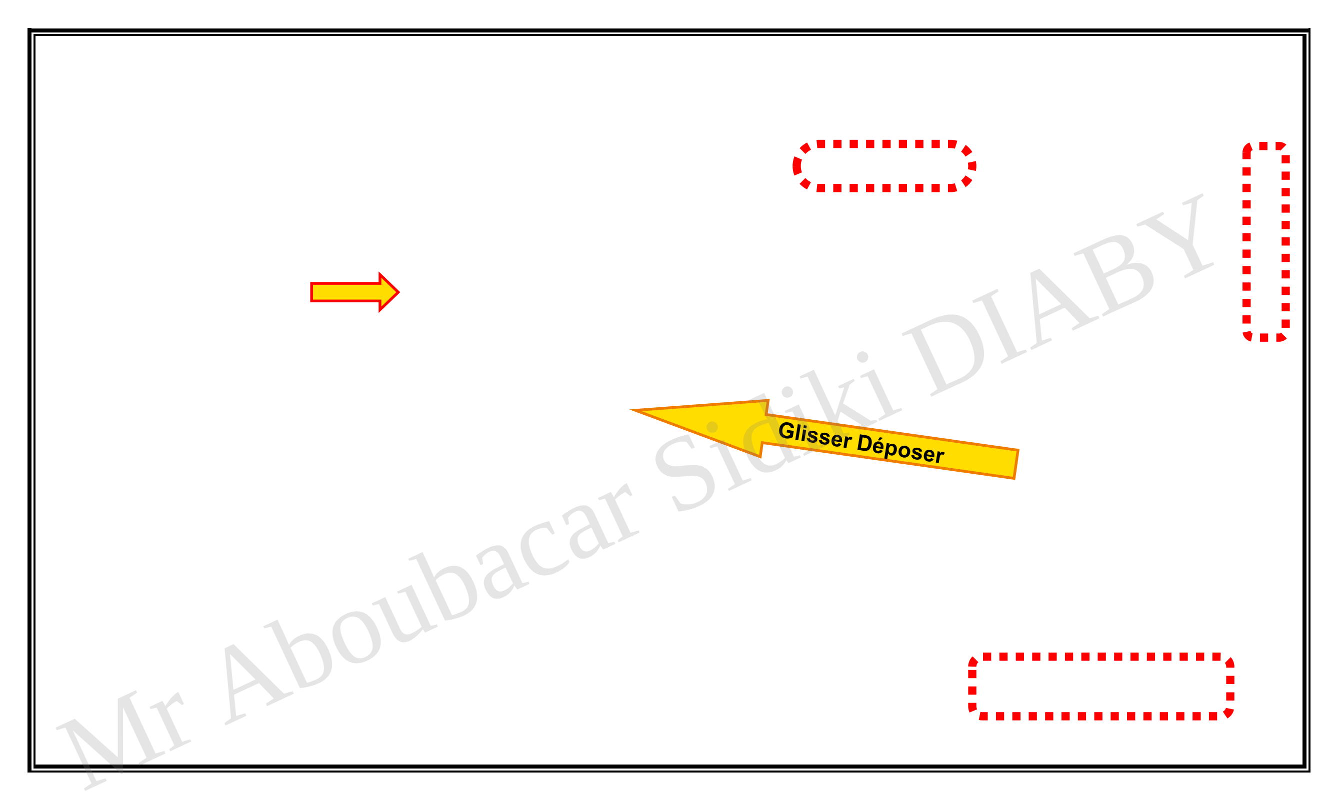
1. Sélectionnez le contrôleur dans le navigateur de projet et renommez le « S7-1500 ».
2. Ouvrez « Vue des appareils » et sélectionnez « Propriétés » dans la fenêtre d'inspection de la CPU.
3. Sous « Fail-safe », désactivez « F-activation ».

   Résultat: le répertoire « Safety Administration » n'est plus visible dans le navigateur de projet.
4. Sous « Protection et Sécurité » activez le niveau de protection « Accès complet, y compris failsafe (pas de protection) » (voir la vue ci dessous).



Enregistrez votre projet.

4.9.6. Exercice 6 : Configurer la station S7-1500



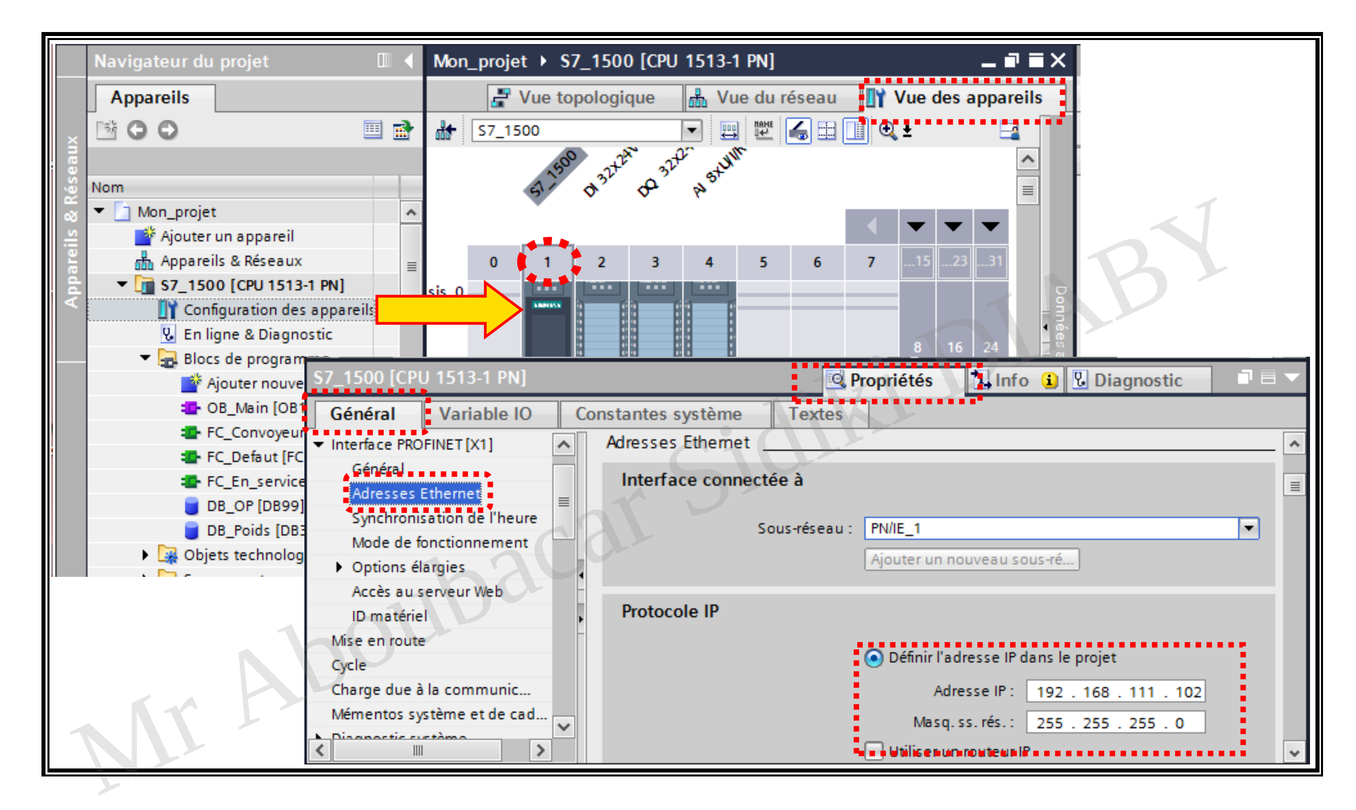
Enoncé

Vous devez configurer la station S7-1500 de votre projet hors ligne de telle façon que la disposition des modules corresponde à votre banc pédagogique.

Marche à suivre

1. Pour les emplacements de 2 à 4, ajoutez les modules E/S par glisser - déposer à partir du « Catalogue du matériel »
   (Dans l'exercice 1 du chapitre « Vue d'ensemble du système », vous avez déjà lu et noté les informations nécessaires sur l'afficheur de la CPU.)
2. Insérez le module d'alimentation dans l'emplacement 1 à partir du catalogue du matériel.
3. Enregistrez votre projet.

4.9.7. Exercice 7 : Propriétés de la CPU: Adresse IP et masque de sous réseau



Enoncé

En mode hors ligne, vous devez attribuer une adresse IP à la CPU S7-1500 du projet.

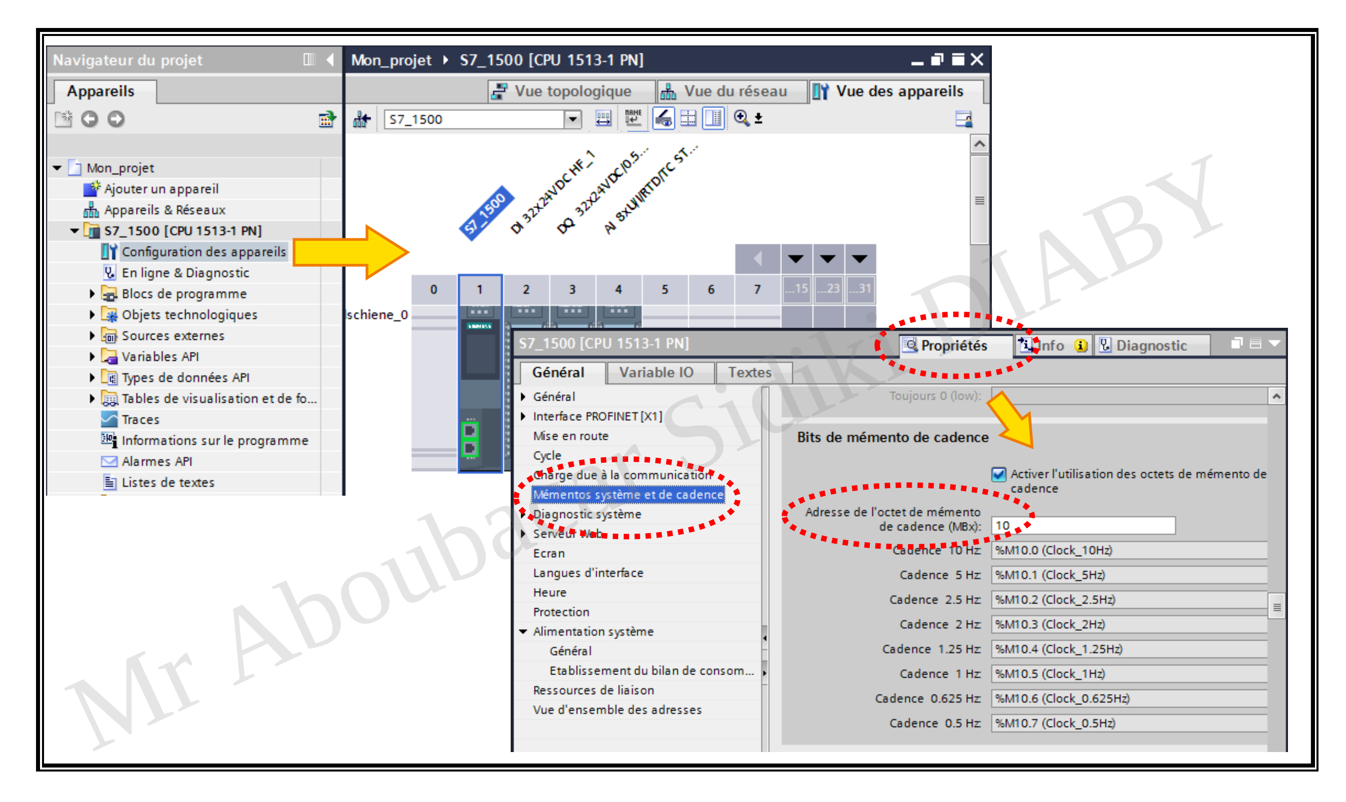


Pour un contrôleur PROFINET-IO il n'est pas indispensable d'attribuer un nom d'appareil. Seuls les périphériques IO (IO-Devices) doivent avoir un nom spécifique pour permettre au contrôleur IO de les identifier.

Marche à suivre

1. Sélectionnez la CPU dans la « Vue des appareils ».
2. Sélectionnez dans la fenêtre d'inspection « Propriétés » et
   - connectez la CPU à un sous-réseau
   - reportez l'adresse IP et le masque de sous réseau indiqués dans la vue.
3. Vérifiez l'affectation d'adresse dans la fenêtre d'inspection sous « Info », sauvegardez votre projet.

4.9.8. Exercice 8 : propriétés de la CPU → contrôler l'octet de mémento de cadence et le fuseau horaire



Enoncé

Activez de l'octet de mémento de cadence dans les propriétés de la CPU → octet de mémento 10.

Marche à suivre

1. Sélectionnez la CPU

Sélectionnez la CPU dans la fenêtre de travail.

Les propriétés de la CPU s'affichent alors dans la fenêtre d'inspection.

2. Activez le mémento de cadence et affectez l'adresse 10

Sélectionnez dans la
>> Fenêtre d'inspection > Onglet « Propriétés » > Rubrique « Mémentos système et de cadence »

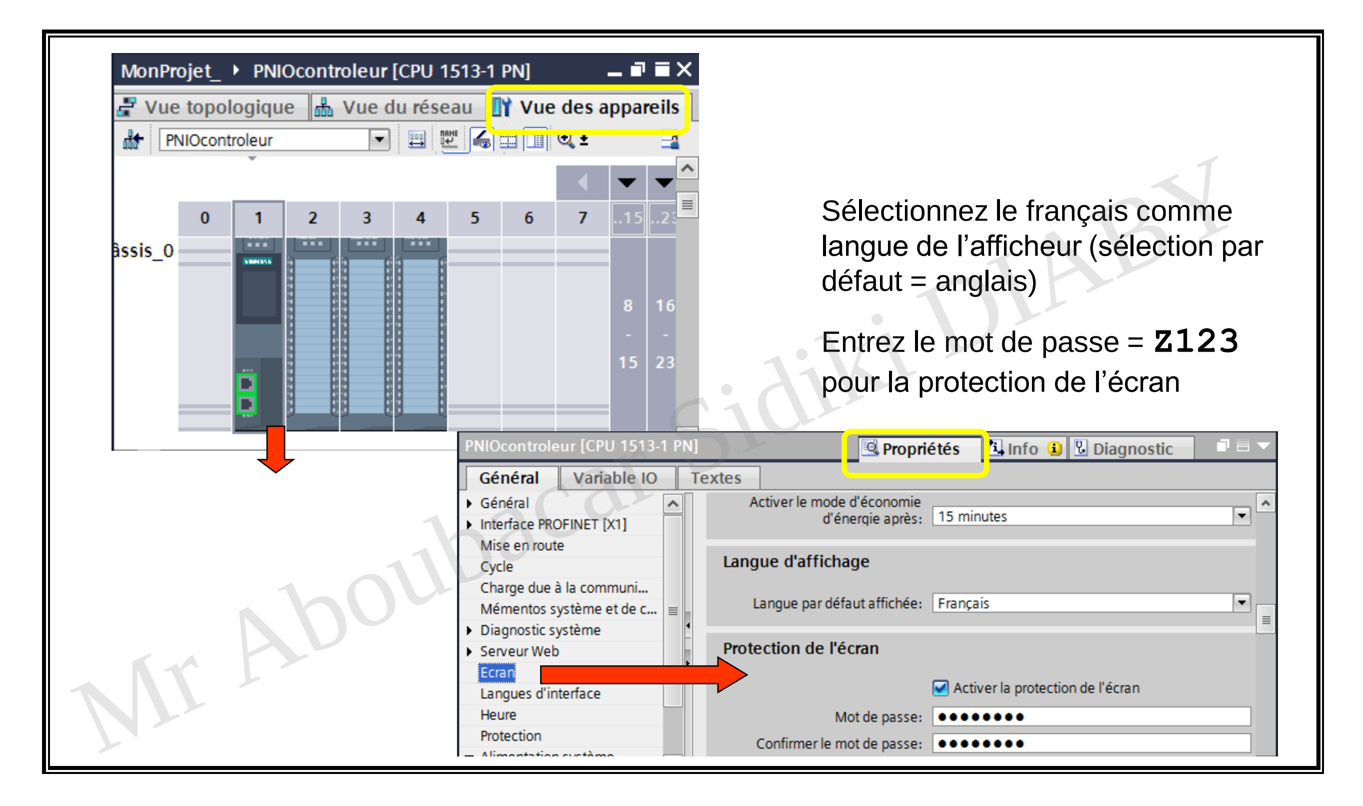
Activez l'octet de mémento de cadence avec l'adresse 10.

3. Paramétrez le fuseau horaire de la CPU

Sélectionnez dans la
>> Fenêtre d'inspection > Onglet « Propriétés » > Rubrique « Heure »

et sélectionnez l'heure locale.

4.9.9. Exercice 9 : propriétés de la CPU → paramétrer le mot de passe de l'afficheur



Enoncé

Modifiez la langue d'affichage de l'afficheur de la CPU dans les propriétés de la CPU et activez la protection de l'afficheur.

Marche à suivre

1. Sélectionnez la CPU

Sélectionnez la CPU dans la fenêtre de travail.

Les propriétés de la CPU s'affichent dans la fenêtre d'inspection.

2. Modifiez les paramétrages de l'afficheur

Sélectionnez dans la
>> *Fenêtre d'inspection > Onglet « Propriétés » > Rubrique « Ecran »*

Sélectionnez comme langue d'affichage : Français

Activez la protection de l'afficheur avec le mot de passe : **Z123**



Le système ne fait pas la distinction entre majuscules et minuscules, car seuls les lettres A..Z et les chiffres 0..9 peuvent être sélectionnés sur l'afficheur.
Comme on ne dispose pas de clavier, il est recommandé de choisir un mot de passe simple (éventuellement entièrement numérique).

3. Enregistrezle projet

Enregistrez votre projet Enregistrer le projet .

4.9.10. Exercice 10 : Module d’entrée TOR - Paramétrer l’adresse du module

Quand et dans quelle mémoire image les entrées seront actualisées.

Enoncé

Paramétrez l’adresse du module d’entrée TOR (adresse de début) comme indiqué dans la figure ci-dessus.

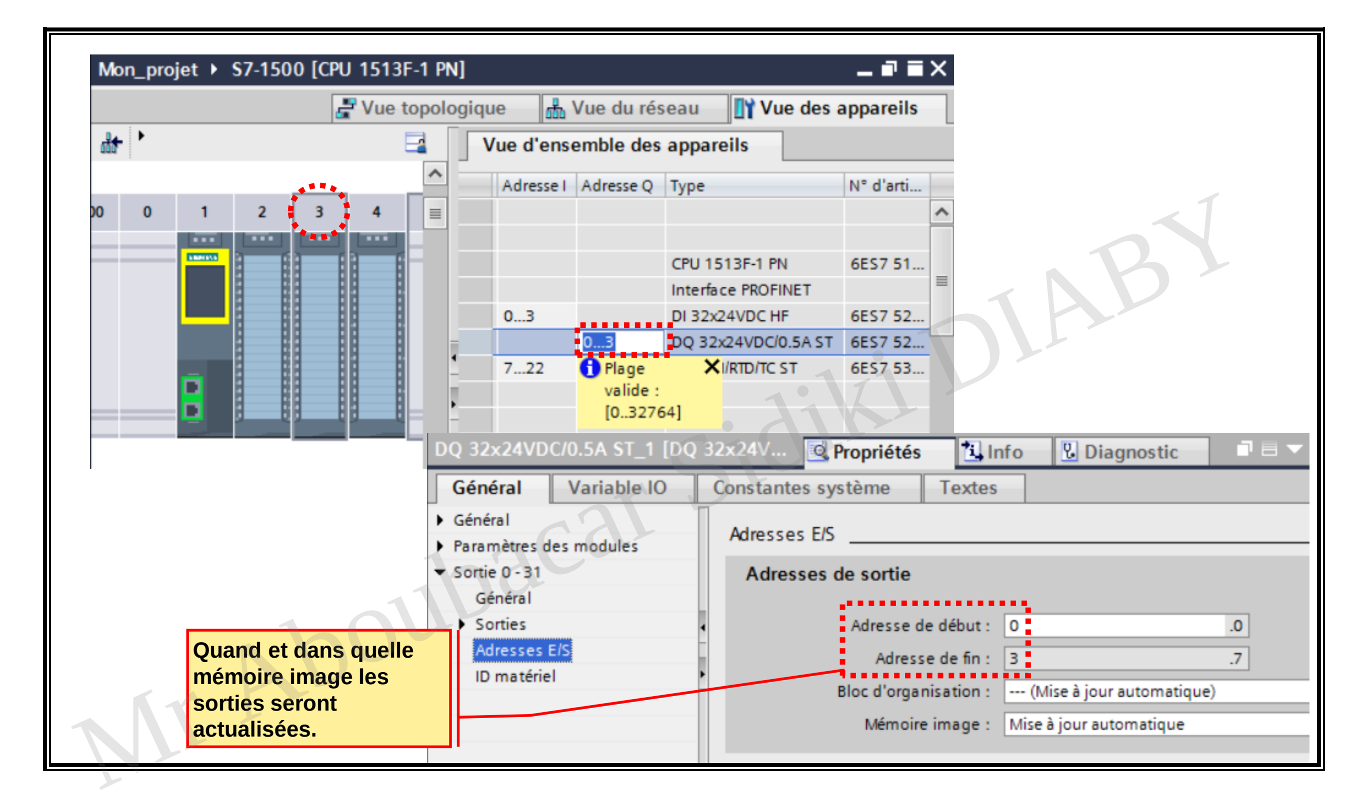
Marche à suivre

1. Sélectionnez le module d’entrée TOR dans la vue des appareils (voir figure).
2. Sélectionnez l’onglet « Propriétés » dans la fenêtre d’inspection, puis « Adresses E/S ».
3. Dans la boîte de dialogue qui apparaît, entrez les adresses indiquées dans la figure ci-dessus.

Notes:

1. La CPU 1500 dispose de 31 mémoires image partielles. Les mémoires image partielle « PIP 1 » à "PIP 31" peuvent être associées à certains blocs d’organisation spécifiques. Après l’exécution de l’OB, la mémoire image partielle pour ces entrées est mise à jour par le système. A la fin de l’OB, les sorties de cette mémoire image partielle sont écrites sur les sorties de périphérie par le système. Les mémoires image partielle sont exclues du rafraîchissement automatique.
2. Une mémoire image partielle peut être mise à jour dans le programme utilisateur avec des instructions spécifiques. Pour cela, il existe les fonctions « UPDAT_PI » pour les mémoires image partielle des entrées et « UPDAT_PO » pour la mémoire image partielle des sorties.

4.9.11. Exercice 11 : Module de sortie TOR - Paramétrer l'adresse du module



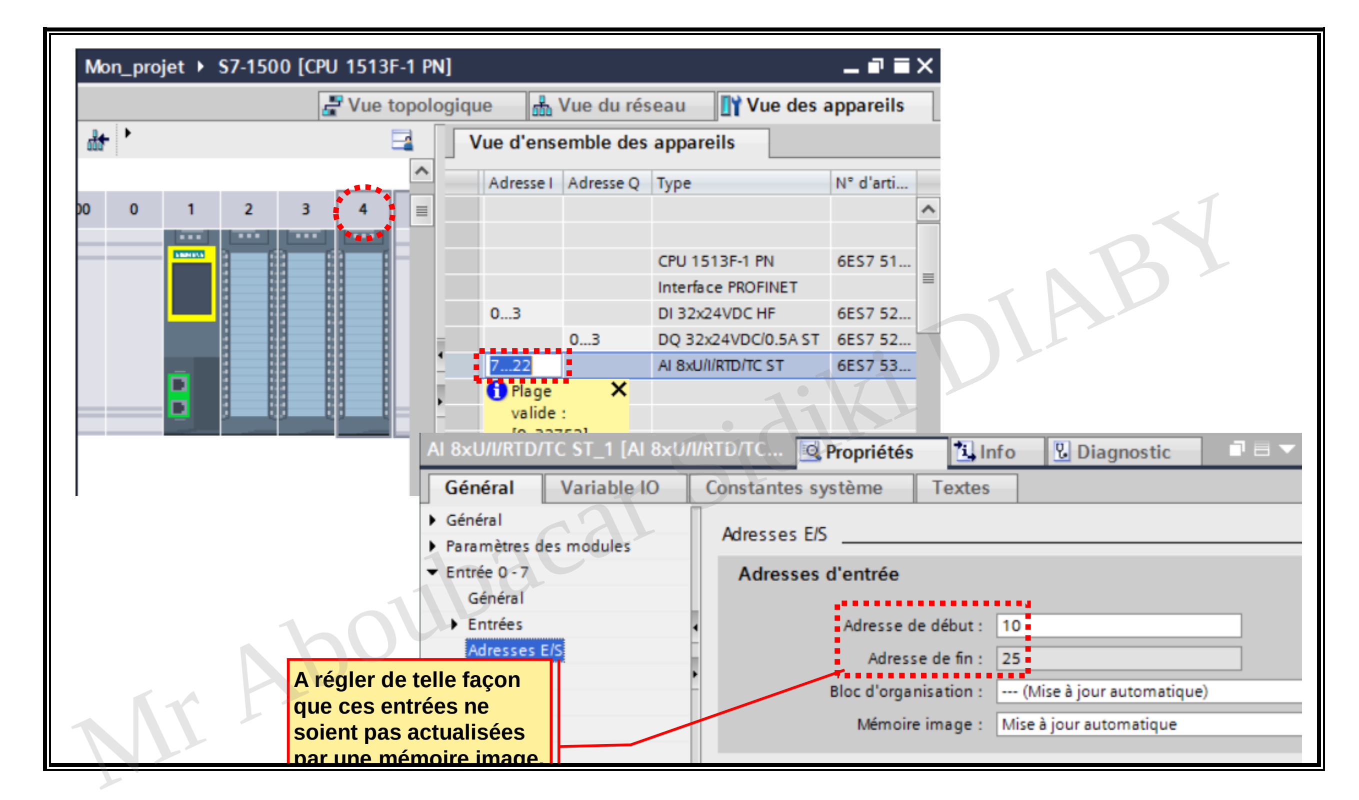
Enoncé

Paramétrez l'adresse du module de sortie TOR (adresse de début) comme indiqué dans la figure ci-dessus.

Marche à suivre

1. Sélectionnez le module d'entrée TOR dans la vue des appareils (voir figure).
2. Sélectionnez l'onglet « Propriétés » dans la fenêtre d'inspection, puis « Adresses E/S ».
3. Dans la boîte de dialogue qui apparaît, entrez les adresses indiquées dans la figure ci-dessus, et retenez « Mise à jour automatique » sous « Mémoire image ».
4. Enregistrez votre projet

4.9.12. Exercice 12 : Adressage du module AI



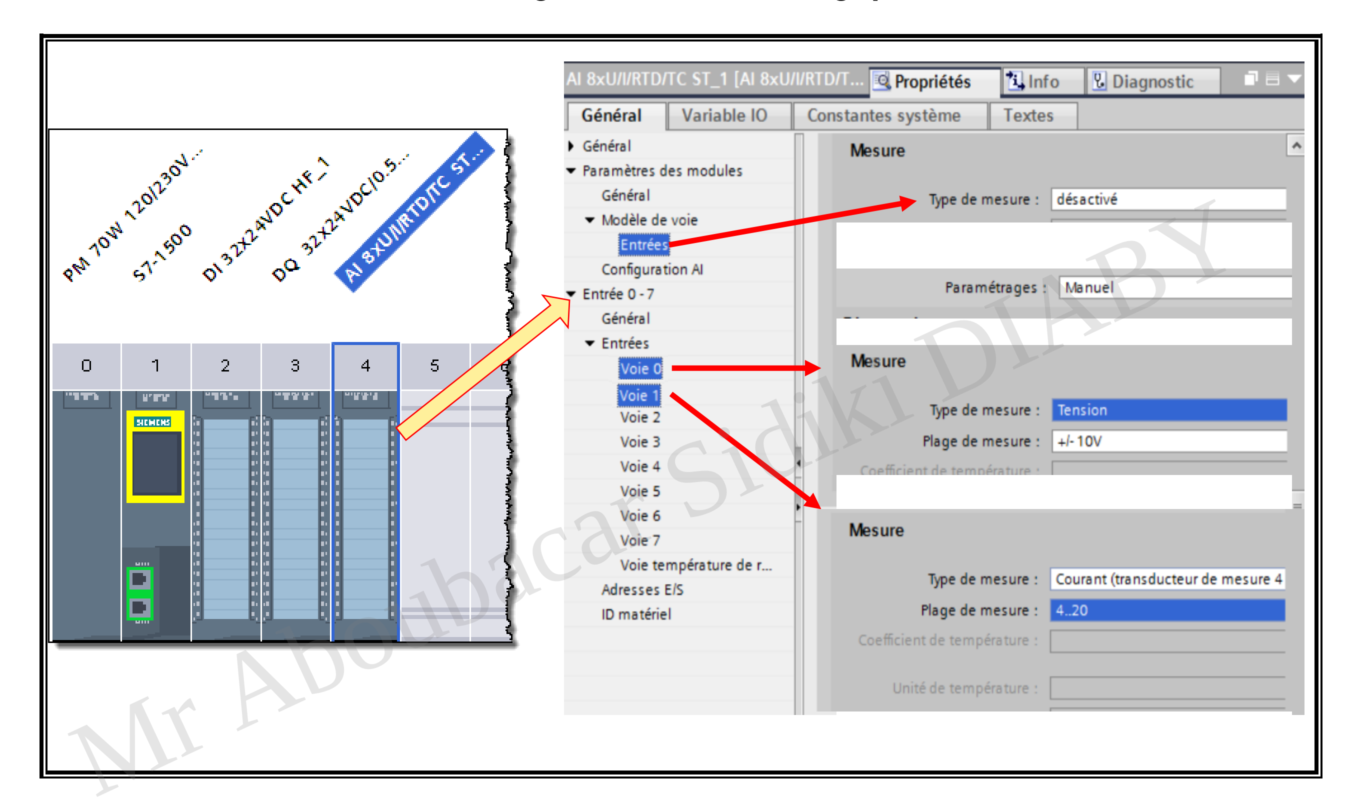
Enoncé

Paramétrez l'adresse du module d'entrée analogique (adresse de début) comme indiqué dans la figure ci-dessus.

Marche à suivre

1. Sélectionnez le module d'entrée AI dans la vue des appareils (voir figure).
2. Sélectionnez l'onglet « Propriétés » dans la fenêtre d'inspection, puis « Adresses E/S ».
3. Dans la boîte de dialogue qui apparaît, entrez les adresses indiquées dans la figure ci-dessus.
4. Régler « Bloc d'organisation » sur « Aucun » car la lecture de la valeur analogique sera directement réalisée par programme ainsi il n'est pas nécessaire de la rafraîchir par une mémoire image.
5. Sauvegardez votre projet.

4.9.13. Exercice 13: Paramétrage du module analogique AI

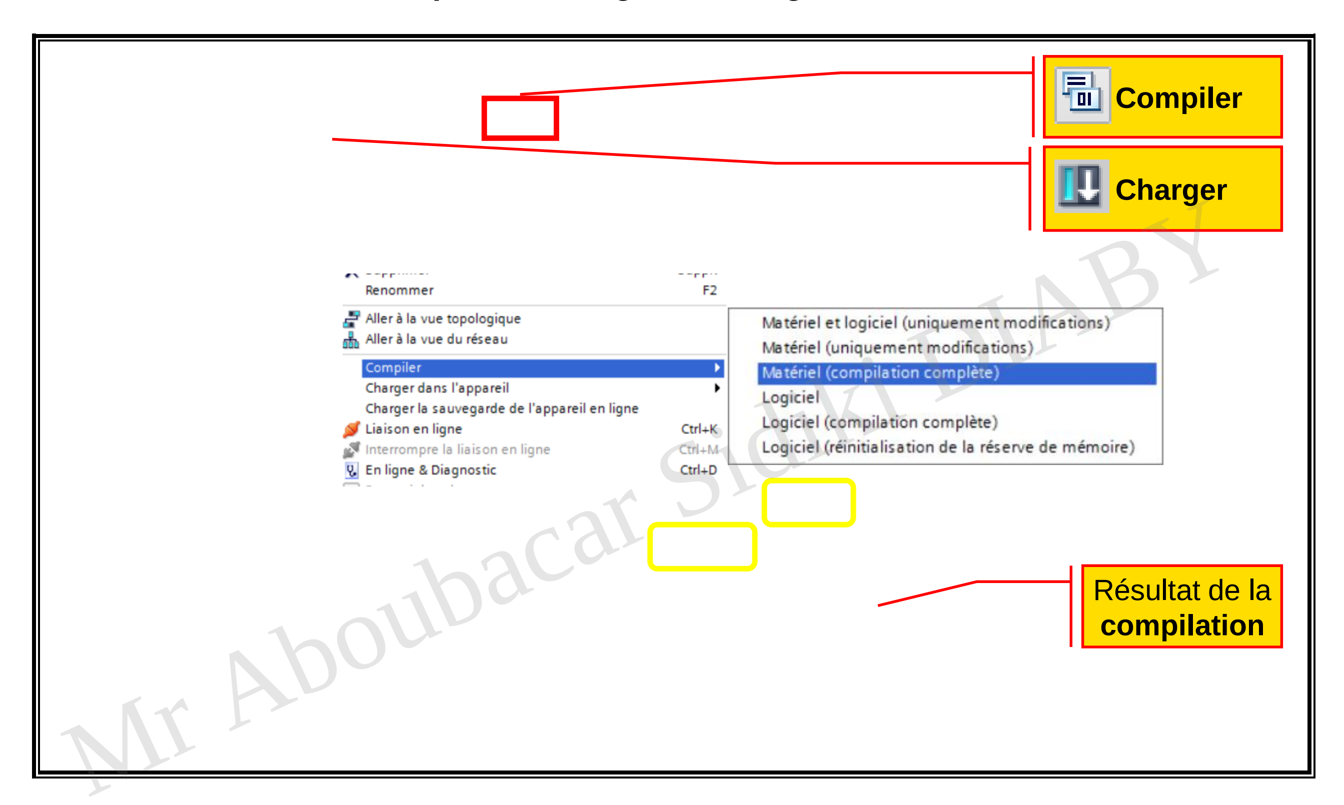


Enoncé:

Réglage des paramètres du module analogique AI

- Réglage de la mesure du modèle de voie sur désactivé
- Voie 0 → Tension +/- 10V
- Voie 1 → Courant (transducteur de mesure 4 fils) 4..20 mA
- Sauvegardez votre projet

4.9.14. Exercice 14 : compiler et charger la configuration matérielle dans la CPU



Enoncé

Vous devez compiler puis charger dans la CPU la configuration et le paramétrage de la station matérielle S7-1500.

Veillez à ne charger que la configuration du matériel (pas de programme) !

Problème

La CPU ne passe en RUN que si au moins une composante de programme (par ex. variable API) est chargée. **La configuration des appareils ne suffit pas à elle seule et un chargement sélectif n'est pas autorisé !**

Marche à suivre

1. Dans le navigateur du projet, sélectionnez votre station S7-1500.
2. Compiler la configuration matérielle (clic droit sur la station, vue ci dessus)
3. Après une compilation sans erreur, charger la configuration matérielle dans la CPU (clic droit sur la station, charger dans l'appareil)



Note: Lors de l'utilisation des boutons à gauche une compilation Delta ou un chargement Delta du répertoire et du sous répertoire sélectionnés dans le navigateur du projet est toujours réalisé. Donc, si une station (CPU) est sélectionnée, la compilation Delta ou le chargement Delta de la station complète, (matérielle et logicielle) est réalisée.

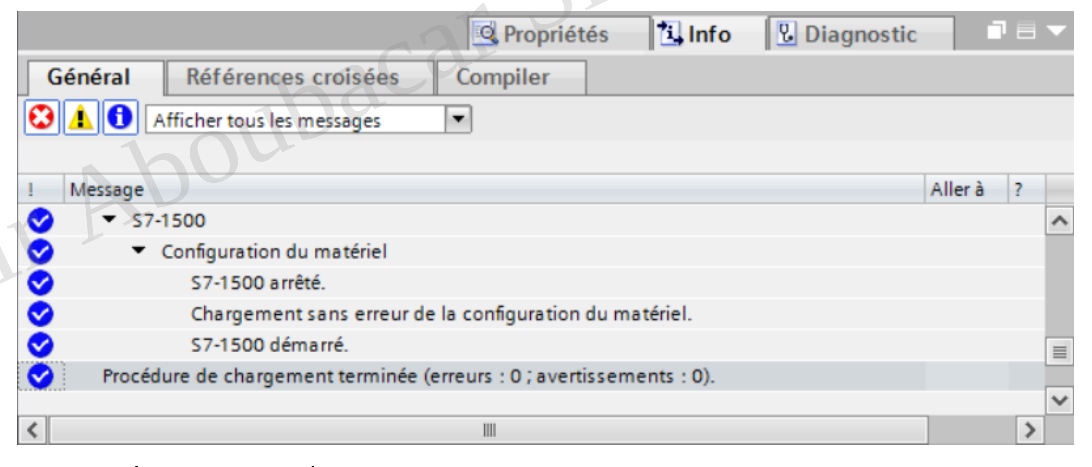
Suite page suivante

4. Confirmez les informations du dialogue par le bouton « charger »:

5. Après l'activation du chargement, un autre dialogue apparaît qui est clôturé par « Terminer ».

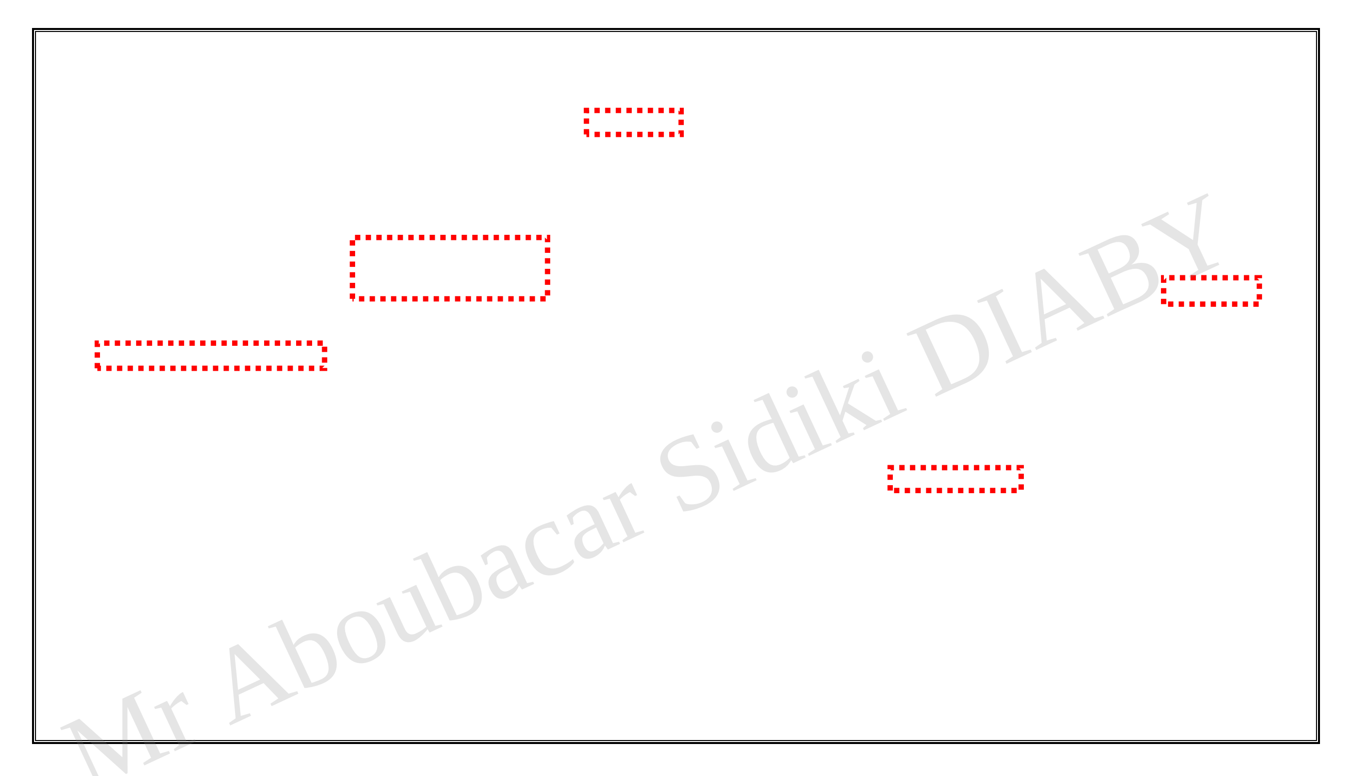
Si vous avez uniquement chargé la configuration matérielle dans la CPU celle-ci devrait se trouver en mode STOP.

6. Dans la fenêtre d'inspection sous « Info » -> « Général » vérifiez le résultat du chargement de la configuration matérielle:



7. Enregistrez votre projet

4.9.15. Exercice 15: Réglage de l'heure et passage en mode RUN du contrôleur



Enoncé

Avec TIA Portal, vous allez régler l'horloge du contrôleur et essayer de le démarrer.

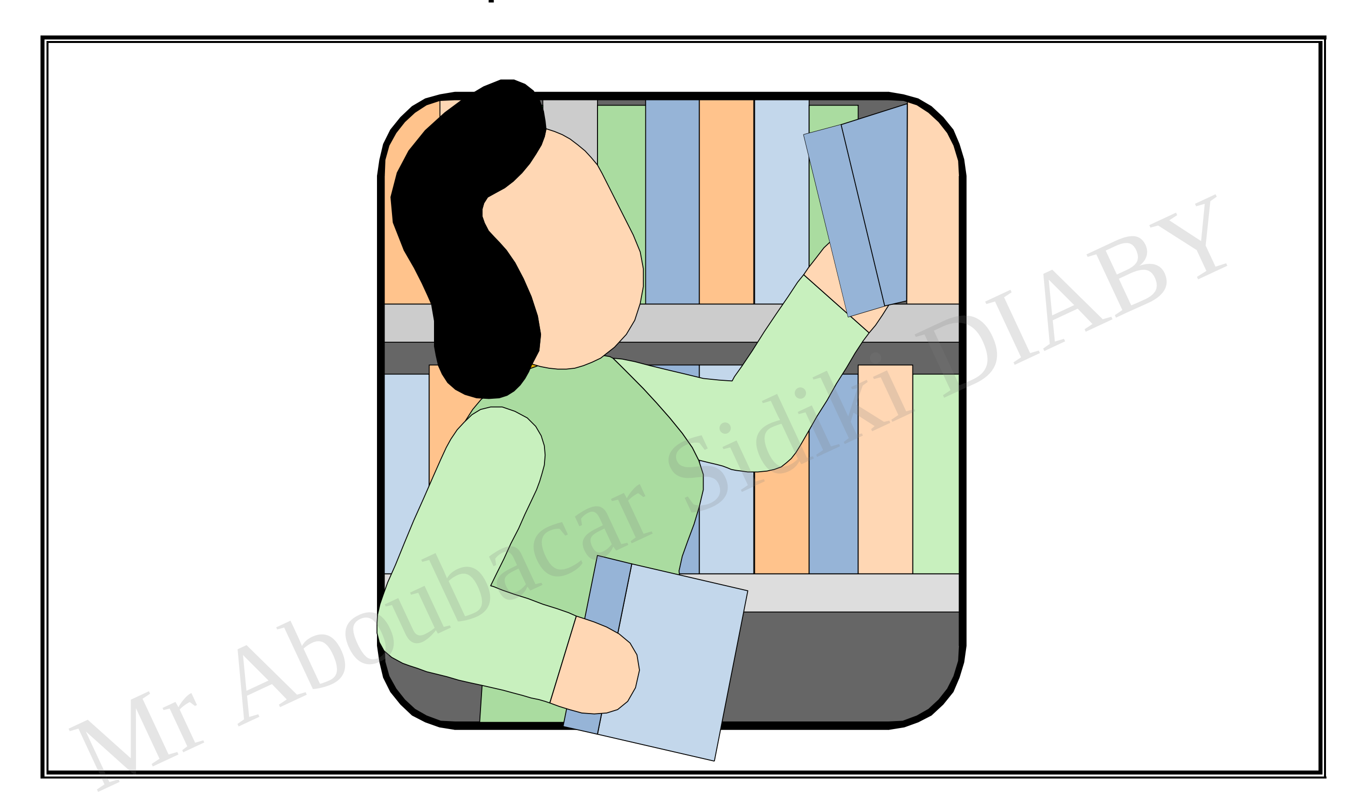
Marche à suivre

1. Accédez à l'accès en ligne du S7-1500 via « En ligne & Diagnostics » dans le navigateur de projet.
2. Etablissez une liaison en ligne avec le contrôleur via « Liaison en ligne » (Le bouton « Liaison en ligne « se trouve dans la barre des tâches et dans la fenêtre de travail sous « Accès en ligne »).
3. Dans la fenêtre « Accès en ligne » sélectionnez « Fonctions > Régler l'heure ». (L'heure de la PG/du PC et l'heure de la CPU sont affichées.)
4. Adoptez l'heure de la PG/du PC par « Adopter l'heure de la PG/du PC » et confirmez par le bouton appliquer.

Remarque:

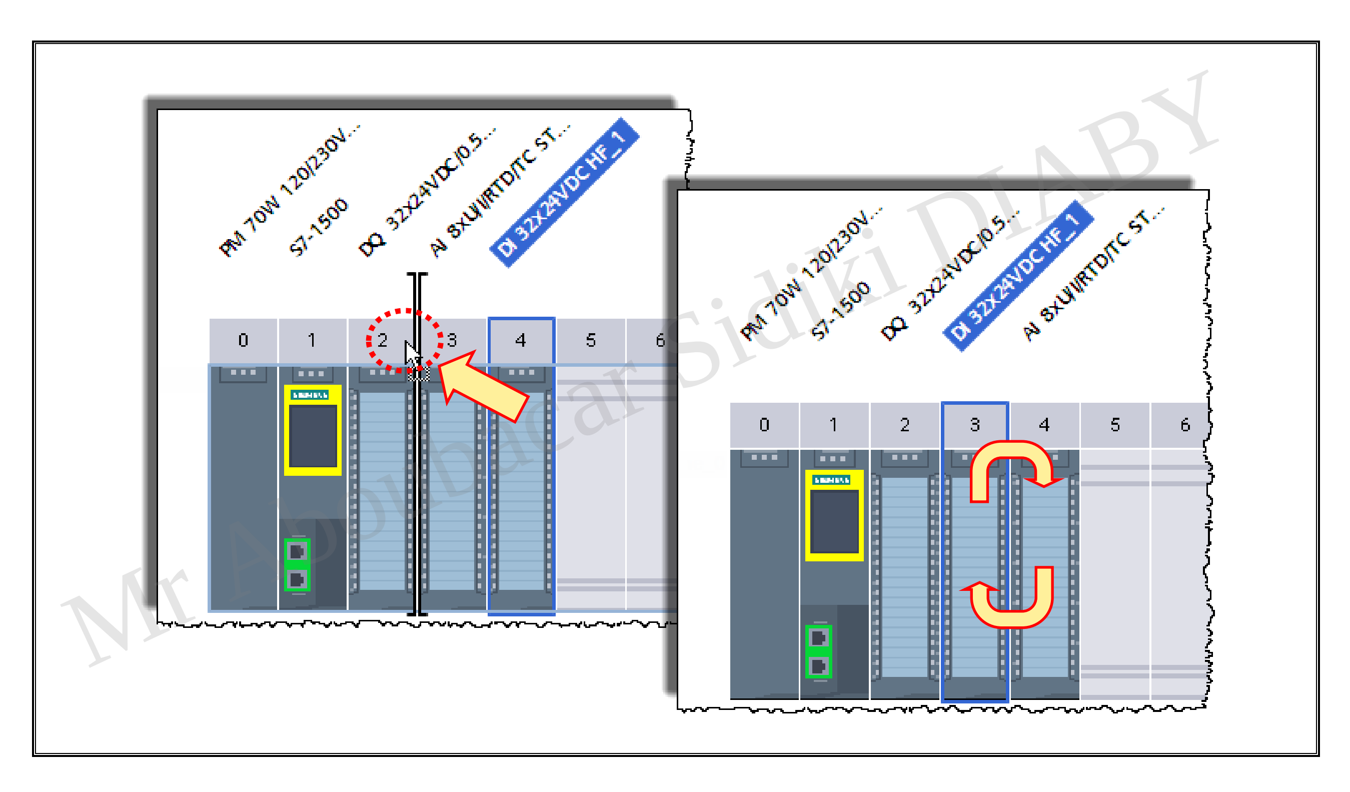
Si « Adopter l'heure de la PG/du PC » est désactivée alors l'horloge de la CPU est modifiable manuellement.

4.10. Informations complémentaires





4.10.1. Inversion de l'emplacement / Insertion d'un module entre 2 autres



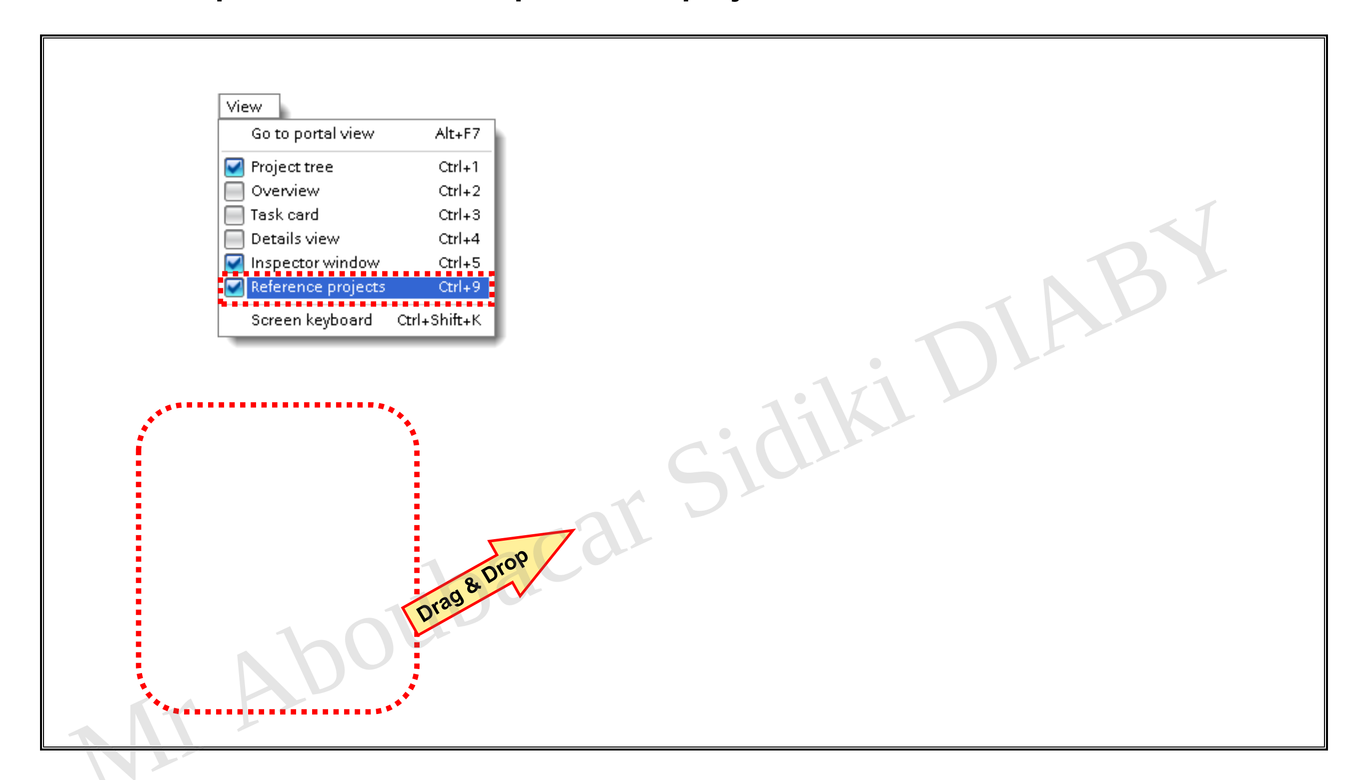
Echanger un emplacement ou insérer un module entre deux modules.

Par Glisser & Déposer déplacer le module avant ou après l'emplacement jusqu'à l'apparition des repères de l'emplacement retenu, comme illustré dans la vue. Le module est inséré au niveau des repères et les modules sont déplacés d'un emplacement vers la droite. S'il reste trop de modules, les modules excédentaires sont déplacés vers la zone « Modules non enfichés ».

Note:

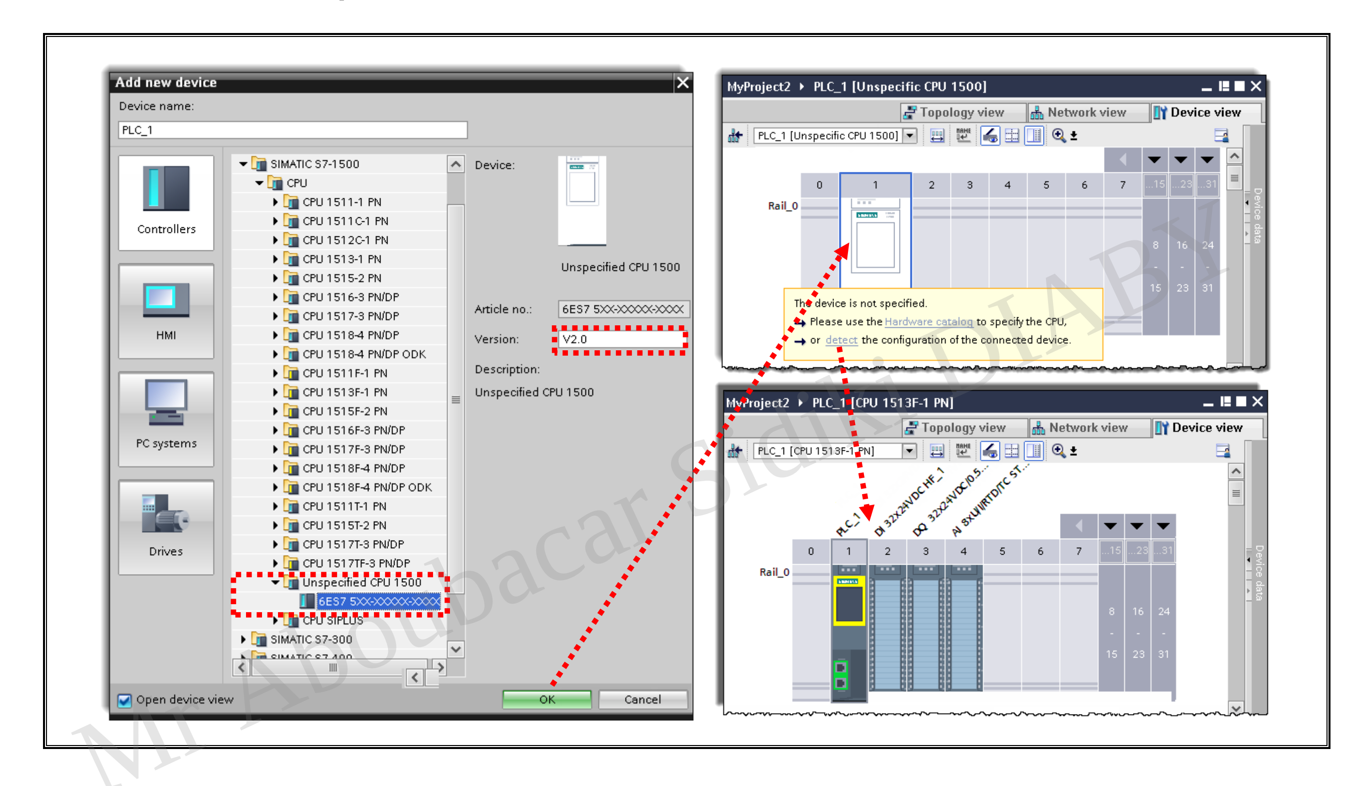
Ceci n'est pas possible à partir du catalogue matériel. La modification est uniquement prévue pour des modules déjà insérés ou issus de la zone des « Modules non enfichés ».

4.10.2. Copier des modules à partir d'un projet de référence



Via le menu « Affichage » sous « Projet de référence » vous accédez à l'ouverture de projets en lecture seule. Les modules peuvent être copiés dans la vue des modules à partir d'un projet de référence. Ainsi, toutes les affectations de paramètres sont reprises.

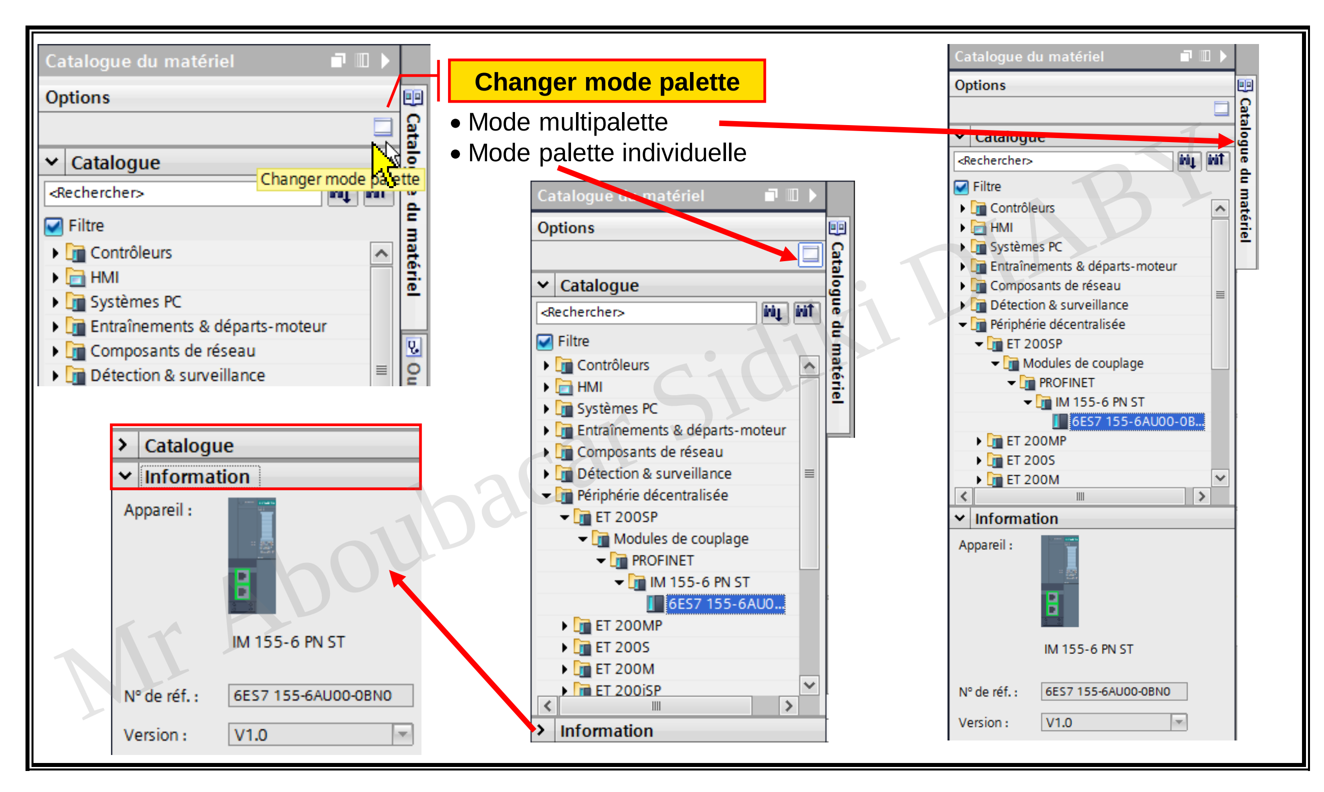
4.10.3. CPU non spécifiée



Lors du choix du contrôleur, vous pouvez également retenir une « CPU non spécifiée ». Ceci peut être nécessaire si vous ne connaissez pas la CPU retenue et que vous voulez commencer à programmer. A ce niveau, vous avez uniquement à spécifier le firmware de la CPU qui sera en place ultérieurement. La version du firmware doit être retenue car quelques fonctions et instructions utilisées pour la programmation dépendent de la version du firmware.

Ensuite vous pouvez programmer sans avoir configuré le matériel. Le matériel peut être configuré ultérieurement à partir du catalogue du matériel ou déterminé en ligne.

4.10.4. Paramètres d'affichage pour la Task Card « Catalogue du matériel »



Vous avez le choix entre deux modes d'affichage :

- Mode palette individuelle:
  Une seule palette peut être ouverte à la fois. Si l'on ouvre une autre palette, la palette actuellement ouverte se ferme automatiquement.
- Mode multi-palette:
  Plusieurs palettes peuvent être ouvertes en même temps.

Paramétrage pour la configuration des appareils

Lors de la configuration des appareils (CPU, modules périphériques), on doit en général choisir entre plusieurs versions d'un module.

Comme cette information complémentaire sur le module sélectionné dans le catalogue s'affiche dans la palette « Information », il recommandé de choisir dans ce cas le **mode multi-palette**.

Sommaire

5

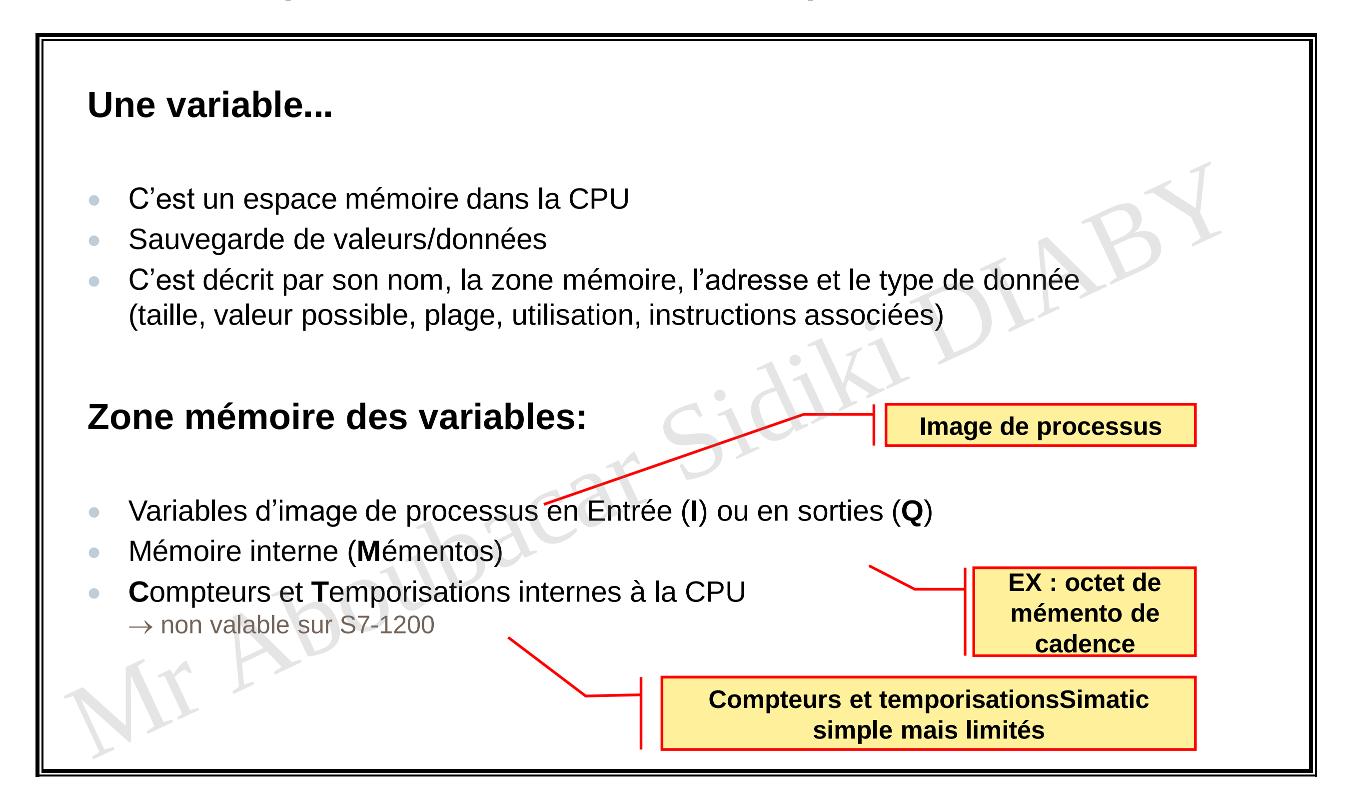
5. Variables API

A l'issue du chapitre, vous allez ...

- ... connaître la signification et l'utilisation des variables
- … connaître la différence entre variables locales et globales
- ... connaître la différence entre l'adressage absolu et l'adressage symbolique
- ... savoir déclarer une table de variables API
- ... savoir renommer et réassigner des variables API
- … savoir visualiser une table des variables



5.1. Que représentent les variables et à quoi servent-t- elles ?



Signification des variables API

Les variables sont, avec les instructions, les éléments les plus importants pour un système de programmation. Leur tâche consiste à enregistrer des valeurs dans un programme afin de pouvoir les traiter ultérieurement.

Types de données

Le type de données détermine les valeurs qui pourront être traitées ainsi que les opérations qui pourront être exécutées avec ces valeurs.

Zones mémoire des entrées et des sorties

Les données sont stockées dans différentes zones de la mémoire d'un contrôleur. Les entrées des modules d'entrées sont stockées dans la mémoire image des entrées où elles sont accessibles en lecture cohérente pendant la totalité du cycle. Le contrôle du processus est réalisé par la mémoire image des sorties qui est écrite sur les modules de sorties.

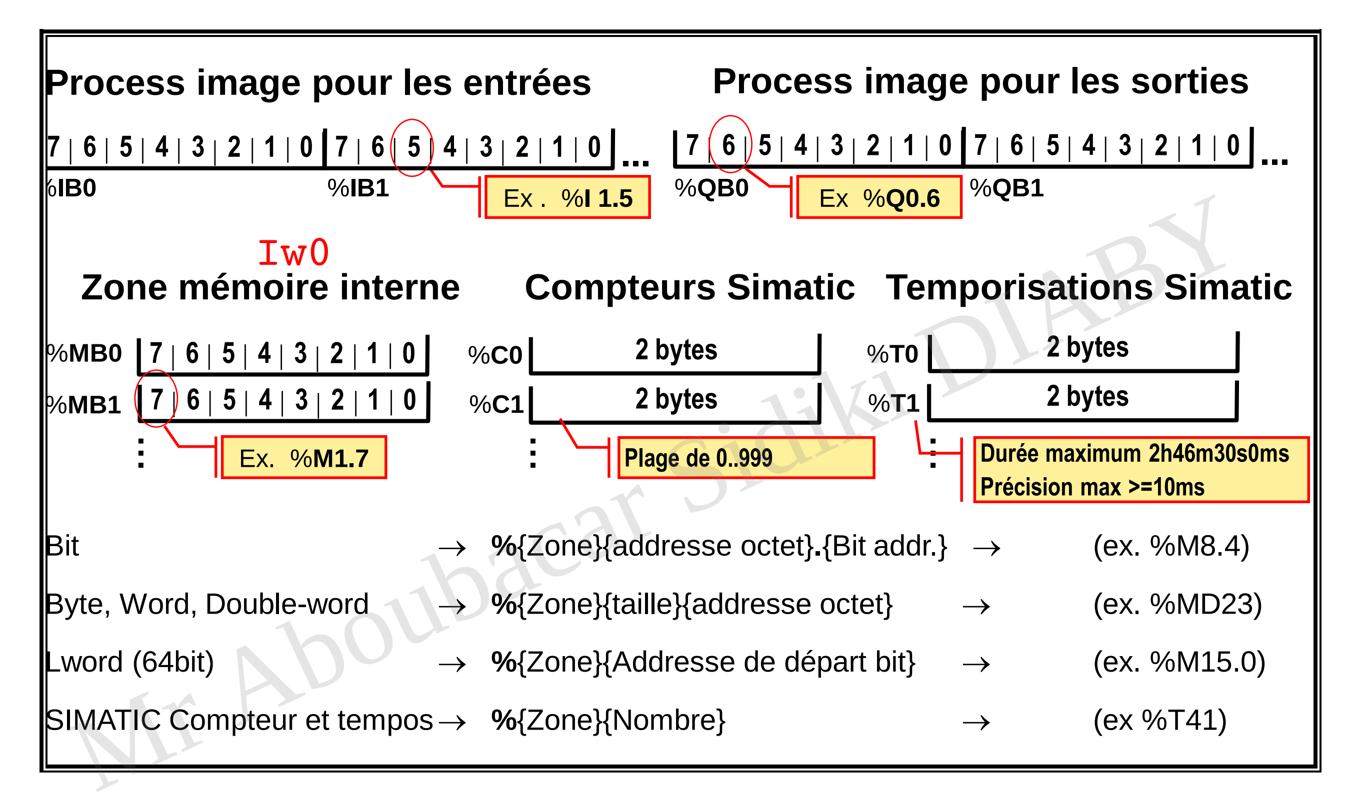
Zone des mémentos

En interne il existe une autre zone de stockage pour enregistrer des valeurs. Mais comme la lisibilité et la réutilisabilité des variables individuelles de la zone des mémentos n'est pas assurée il existe d'autres zones qui seront traitées dans les chapitres suivants

Zone de comptage et temporisation propre à la CPU (Compteur et temporisation Simatic)

Ils représentent une manière simple de réaliser des fonctions de comptage et de temporisation. Les compteurs et temporisations SIMATIC restent cependant très limitées. Les compteurs disposent d'une plage de valeurs allant de 0 à 999 et les temporisations peuvent traiter des valeurs jusqu'à 2 heures 46 minutes et 30 secondes avec une précision de 10 millisecondes et ne sont pas disponibles pour les CPU du S7-1200.
Pour des applications de temporisation et de comptage plus conséquentes, il existe des fonctions qui seront également développées ultérieurement.

5.1.1. Adressage des variables API



Adressage

L'adressage permet de spécifier l'endroit retenu pour l'écriture ou la lecture des variables.

L'adressage commence par l'adresse d'octet numéro « 0 »
Les adresses s'incrémentent en continu, à l'intérieur d'un octet l'adresse bit 0..7 est numérotée de la droite vers la gauche.

Ecriture de l'adresse avec ou sans %

Le caractère % marque la représentation d'une adresse. Il peut être ignoré lors de la saisie de l'adresse car il est automatiquement renseigné par le logiciel.

Variables API > 32 bits:

Pour les variables de taille supérieure à 32 bits (par ex. LWORD), il s'agit, pour l'adressage, de préciser uniquement le bit de départ car l'accès n'est réalisable qu'en mode symbolique. Le nombre d'octets qui doivent être réservés pour ces variables dépend du type de données de ces variables API.

5.2. Types de données élémentaires: binaire et numérique

	Description	Taille(Bit)	S7-1200	S7-1500	Exemple
Types de données binaire	BOOL BYTE WORD DWORD	1 8 16 32	✓	✓	TRUE B#16#F5 W#16#F0F0 DW#16#F0F0FF0F
	LWORD	64	✗	✓	LW#16#5F52DE8B
Types de données numériques	SINT USINT	8	✓	✓	50 20
	INT UINT	16			-23 64530
	DINT UDINT REAL	32			DINT# -2133548520 UDINT#435676 1.0
	LREAL	64			LREAL#-1.0e-5
	LINT ULINT	64	✗	✓	LINT#1543258759 ULINT#154316159

BOOL, BYTE, WORD

Les variables de type de données BOOL représente un bit. Les variables de types de données BYTE ou WORD sont des séquences de 8 ou 16 bits. Les bits ne sont pas évalués individuellement dans ces types de données. Les nombres BCD et la valeur de comptage utilisée en liaison avec une fonction de comptage représentent une forme particulière de ces types de données.

Dans le tableau ci dessus, ***#16#*** traduit une représentation hexadécimale de la valeur.

INT, REAL

Les variables de ces types représentent des nombres avec lesquels il est possible d'effectuer des opérations arithmétiques. (INT –> entier, REAL -> nombre en virgule flottante)

Extensions pour INT, REAL et WORD

U – Non signé

Une variable avec l'extension « U » représente la variable non signée pour le type de donnée correspondant. **Types de donnés:** USINT, UINT, ULINT, UDINT

S - Short

Une variable avec l'extension « S » représente une variable de longueur 8 bits du type de données correspondant. **Types de données:** SINT, USINT

D - Double

Une variable avec l'extension « D » représente une variable de longueur 32 du type de données correspondant. **Types de données:** DWORD, DINT, UDINT

L- Long

Une variable avec l'extension « L » représente une variable de longueur de 64 bits du type de données correspondant. **Types de données:** LWORD, LINT, ULINT, LREAL

5.2.1. Types de données élémentaires : Date, Time et caractère

	Description	Taille(Bit)	S7-1200	S7-1500	Exemple
Type Time	TIME DATE TIME_OF_DAY	32 16 32	✓	✓	T#2h46m30s630ms D#1994-01-21 TOD#18:15:18:999
	S5TIME	16	✗	✓	S5T#1h24m10s
	LTime LTIME_OF_DAY LDT (DATE_AND_LTIME)	64			LT#11350d20h25m14s830ms652µs315ns LTOD#10:20:30.400_365_215 LDT#2008-10-25-08:12:34.567
Type de caractères	CHAR WCHAR	8 16	✓	✓	'R' WCHAR#'w'

TIME, LTIME

Une variable de type TIME (durée en [ms]) occupe un double mot. Cette variable est utilisée, par exemple, pour définir des valeurs de temps des temporisations CEI. Le contenu de la variable est interprété comme un nombre DINT en millisecondes et peut ainsi être positif ou négatif (par exemple: T#1s=L#1000, T#24d20h31m23s647ms = L#2147486470).

Comme TIME, LTIME représente une durée avec une résolution en nanosecondes sur 64 bits. Cela implique, en comparaison avec le type TIME, que des durées avec une résolution plus importante peuvent être enregistrées avec la variable de type LTIME.

DATE

Une variable de type de données DATE est stockée dans un mot sous la forme d'un nombre entier non signé. Le contenu des variables correspond au nombre de jours depuis le 01.01.1990.

TIME_OF_DAY, LTIME_OF_DAY

Le type TOD (TIME_OF_DAY) occupe un double mot et enregistre les millisecondes depuis le début du jour (0:00 heure) sous un entier non signé. Les variables LTOD occupent deux doubles mots et représentent le nombre de nanosecondes depuis le début du jour.

LDT (Date_AND_LTIME)

Le type de données LDT (DATE_AND_LTIME) occupe 8 octets et enregistre les nanosecondes écoulées depuis le 01.01.1970 0:00.

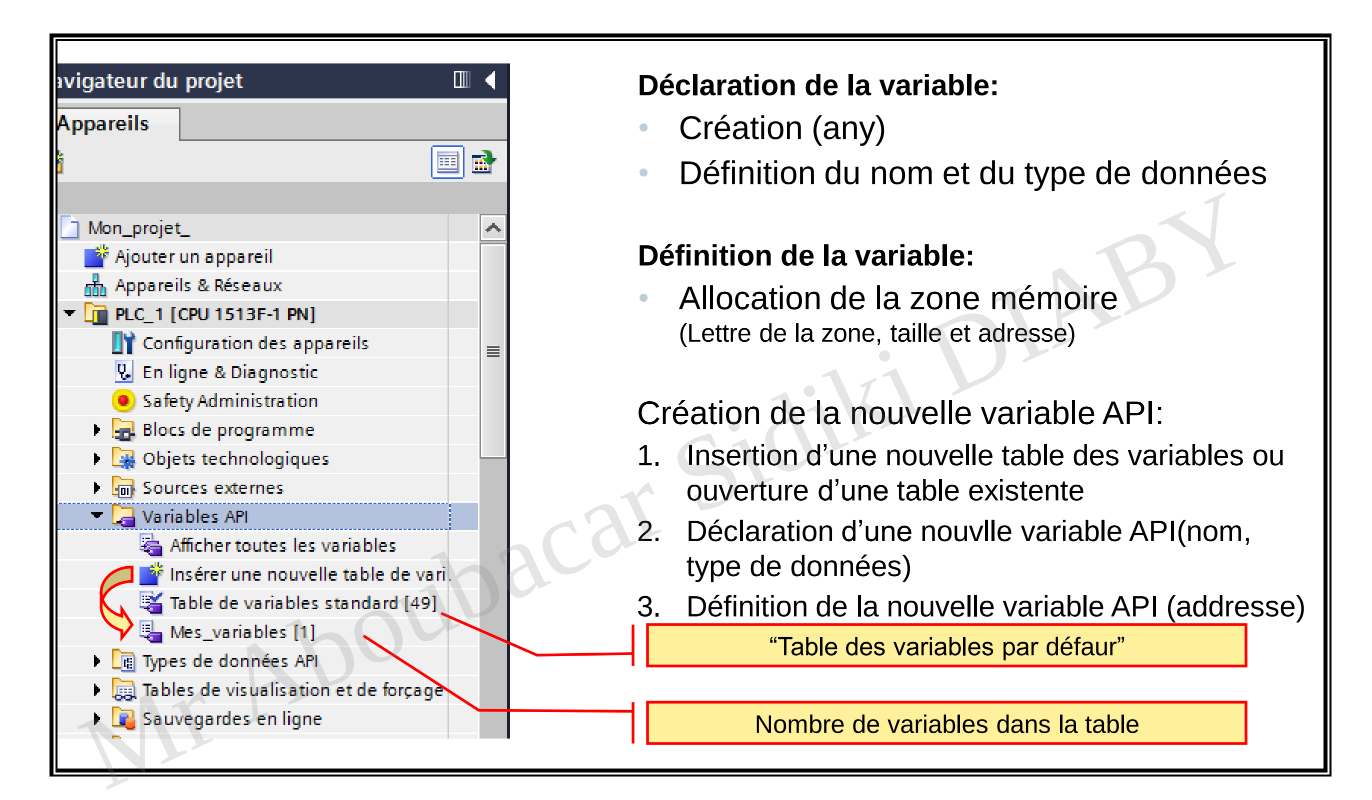
S5TIME

Les variables du type S5TIME définissent les valeurs des temporisations (temporisation S5). Vous indiquez la durée en heures, minutes, secondes ou millisecondes. Vous pouvez saisir la valeur de temporisation avec ou sans caractère de soulignement (1h_4m) ou (1h4m).

CHAR, WCHAR

Le type de données CHAR (8 bits) représente un caractère ASCII et WCHAR (16 bits) un caractère en format Unicode.

5.3. Déclaration et définition des variables API



Tables de variables API

Afin d'obtenir une bonne lisibilité du programme de la CPU, il est judicieux de structurer le stockage des données. Pour cela, on utilise des tables de variables API.

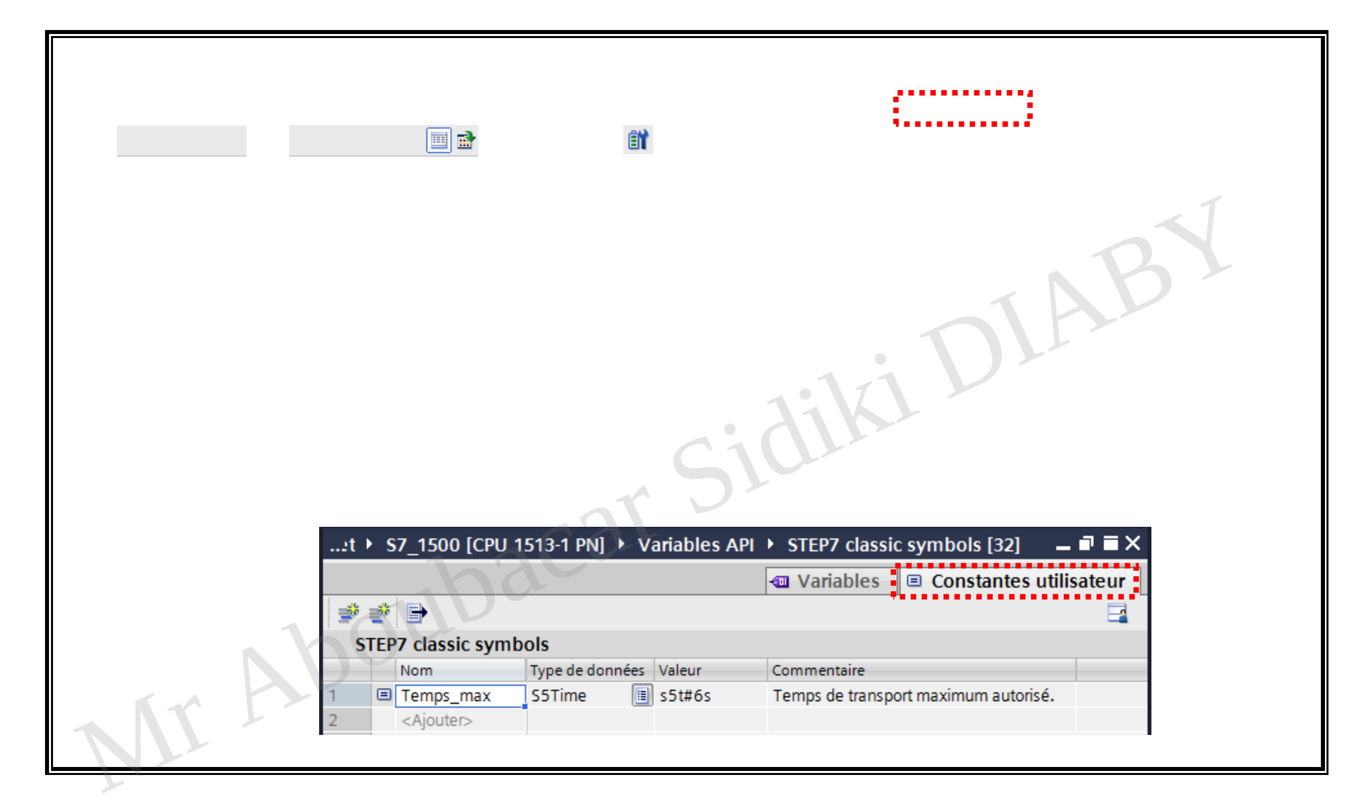
Tables de variables

La table de variables standard (son nom est modifiable) contient des informations CPU supplémentaires et ne peut donc être effacée. Elle peut être utilisée pour n'importe quelle variable API mais pour plus de clarté il est recommandé de définir plusieurs tables de variables.

Déclaration et définition des variables API

Lors de la déclaration d'une variable API dans la table, on définit un nom symbolique (par ex. « "M_Jog_Droite »), le type de donnée (par ex. Bool) et l'adressage absolu (par ex. M16.2).

Variables et constantes API



Variables API

La table de variables API contient la déclaration des variables et constantes valables sur l'ensemble de la CPU et donc globales. Une table de variables API est automatiquement créée pour chaque CPU du projet. La table de variables API contient un onglet pour les variables et un onglet pour les constantes utilisateur, la table de variables standard contient en plus un onglet pour les constantes système. Les variables sont des opérandes avec contenu modifiable utilisés par le programme.

Constantes utilisateur

Une constante définit une valeur invariable d'une donnée. Les constantes sont accédées en lecture pendant l'exécution du programme, mais elles ne peuvent être modifiées. Il n'est pas possible de modifier la valeur d'une constante lors de l'exécution d'un programme.

Dans TIA Portal il est possible d'affecter un nom symbolique aux constantes et ainsi rendre disponibles des variables statiques par les noms. Ces constantes symboliques sont valides pour toute la CPU. La déclaration des constantes s'effectue sous « Constantes utilisateur » de la table de variables API. Les constantes sont des opérandes à contenu invariable et ne nécessitent pas d'adresse absolue.

Créer des variables ou constantes avec la fonction de masse

Vous pouvez créer automatiquement des variables en cliquant sur le symbole de remplissage dans le coin inférieur droit de la cellule, puis en faisant glisser le pointeur vers le bas (cf. Excel).

La création automatique des variables et constantes est possible à l'aide des colonnes « Nom » et « Adresse » (uniquement pour les variables). Les nouvelles variables sont alors créées avec les noms des variables actuelles complétés par un numéro d'ordre. A partir d'une variable nommée par ex. « T_Station », de nouvelles variables nommées « T_Station_1 », « T_Station_2 », etc. sont alors créées.

Constantes système

Les constantes système sont des constantes globales nécessaires au système et créées de manière automatique. Elles sont utilisées par exemple pour l'adressage et l'identification d'objets matériels.

Les constantes système sont attribuées automatiquement lors de l'insertion de modules au niveau des vues des appareils ou du réseau (« Constantes système »). Une constante système est créée pour chaque module, mais également pour chaque « submodul », ainsi un compteur intégré par exemple dispose également d'une constante système. Les constantes système se composent d'un nom symbolique et d'un identifiant numérique du matériel.

Noms des Constantes système

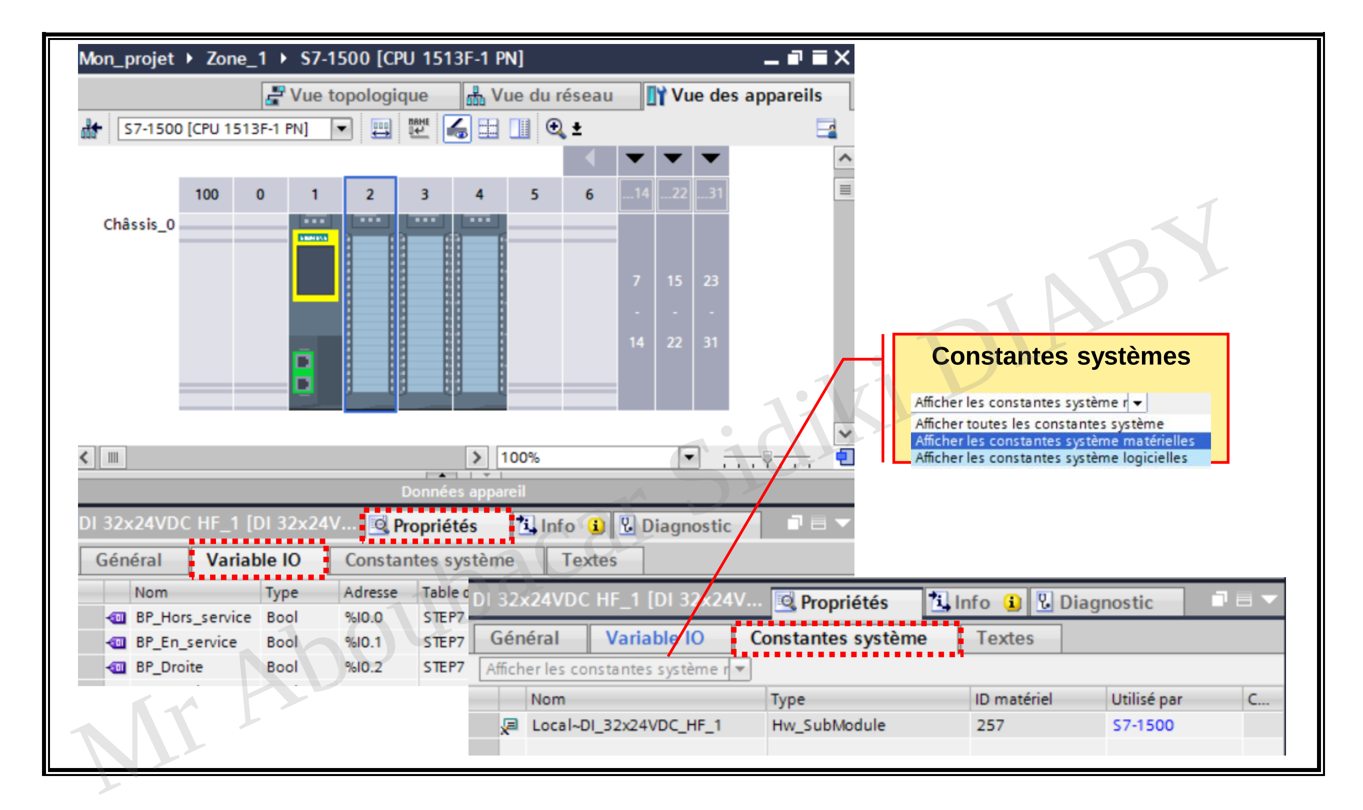
Les noms des Constantes système sont construits de façon hiérarchique. Ils comprennent au maximum quatre niveaux d'hiérarchie séparés par le caractère « ~ ». Il est ainsi possible de reconnaître le « chemin » pour accéder à un module matériel donné.

		Nom	Type de données	Valeur	Commentaire
35		S7_1500[MC]	Hw_SubModule	51	
36		S7_1500	Hw_SubModule	50	
37		S7_1500[Display]	Hw_SubModule	54	
38		S7_1500[Exec]	Hw_SubModule	52	
39		PROFINET-Schnittstelle_1	Hw_Interface	64	
40		Port_1[PN]	Hw_Interface	65	
41		Port_2[PN]	Hw_Interface	66	
42		DI_32x24VDC_HF_1[DI]	Hw_SubModule	258	
43		DQ_32x24VDC_0.5A_ST_1[DO]	Hw_SubModule	259	
44		AI_8xU_I_RTD_TC_ST_1[AI]	Hw_SubModule	260	
45		PROFINET_IO-System[IOSystem]	Hw_IoSystem	265	

Exemple

La constante système portant le nom « Local~PROFINET_Interface_1~Port_1 » représente le port1 de l'interface PROFINET de la CPU

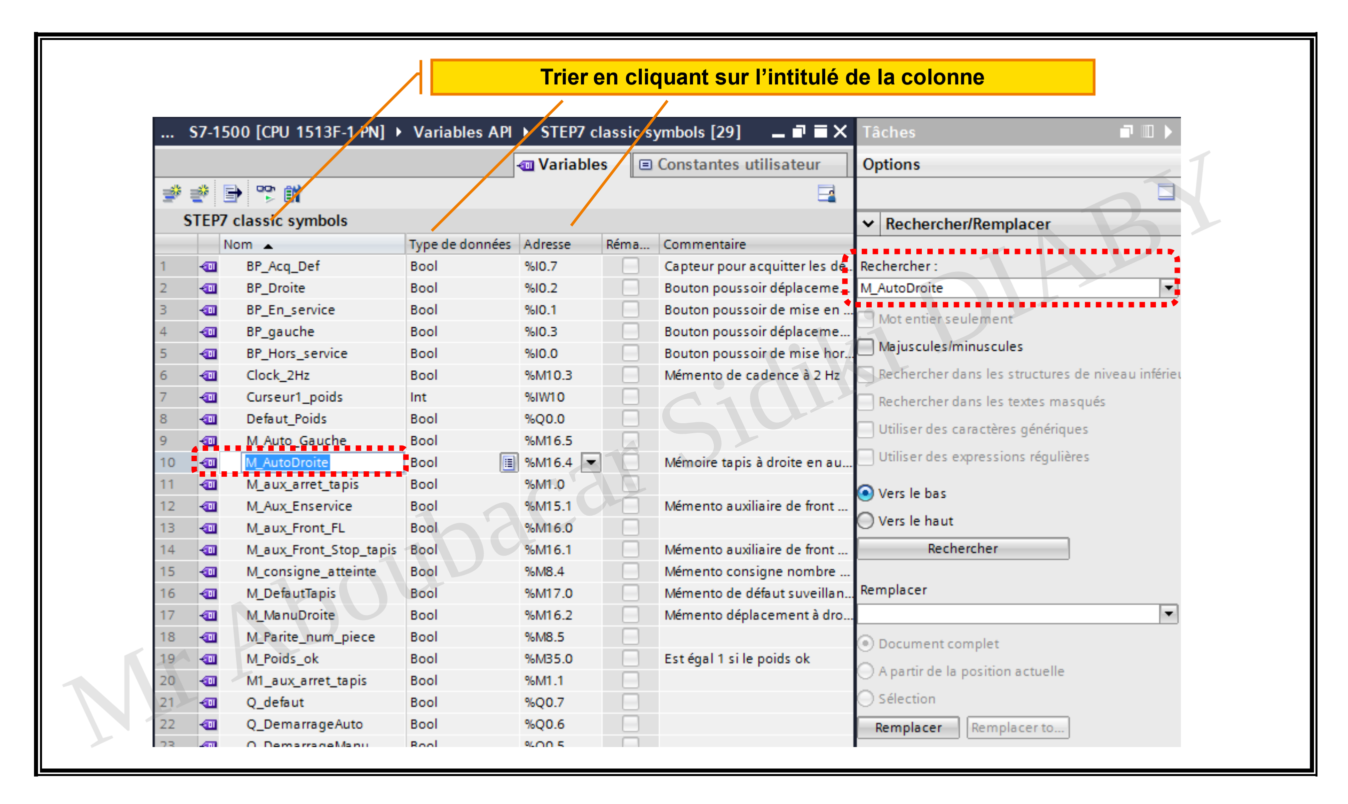
5.3.1. Variables API dans la vue des appareils



Les variables des entrées et des sorties API peuvent également être déclarées et modifiées dans la « Vue des appareils ». Pour cela il faut accéder à « Variables I/O » dans la fenêtre d'inspection des propriétés de cette vue.

Les constantes système du matériel sélectionné sont également accessibles.

5.3.2. Rechercher / remplacer / trier des variables API



Rechercher/Remplacer

La Task Card « Tâches » vous permet de rechercher et de remplacer des variables dans la table de variables API. Vous pouvez également utiliser des caractères génériques dans les critères de recherche (? pour un caractère, * pour plusieurs caractères).

Exemple pour « Rechercher / Remplacer » :

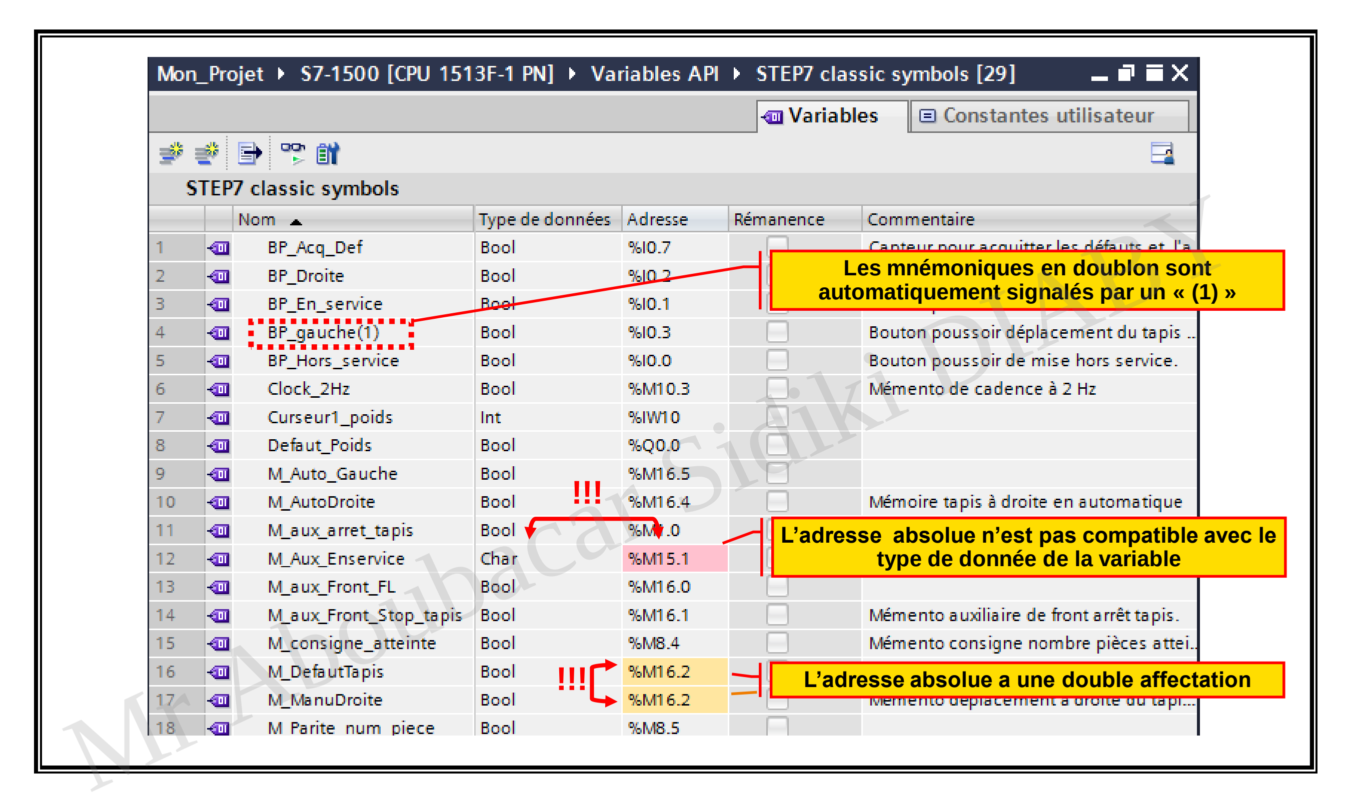
Affecter l'adresse d'octet 4 à toutes les sorties avec l'adresse d'octet 8 :

Rechercher : Q 8.   Remplacer par : Q 4.

Trier

En cliquant sur l'un des intitulés de colonne « Nom », « Type de données » ou « Adresse », les variables sont triées, selon la colonne, par ordre alphabétique ou par adresse (par ordre croissant ou décroissant).

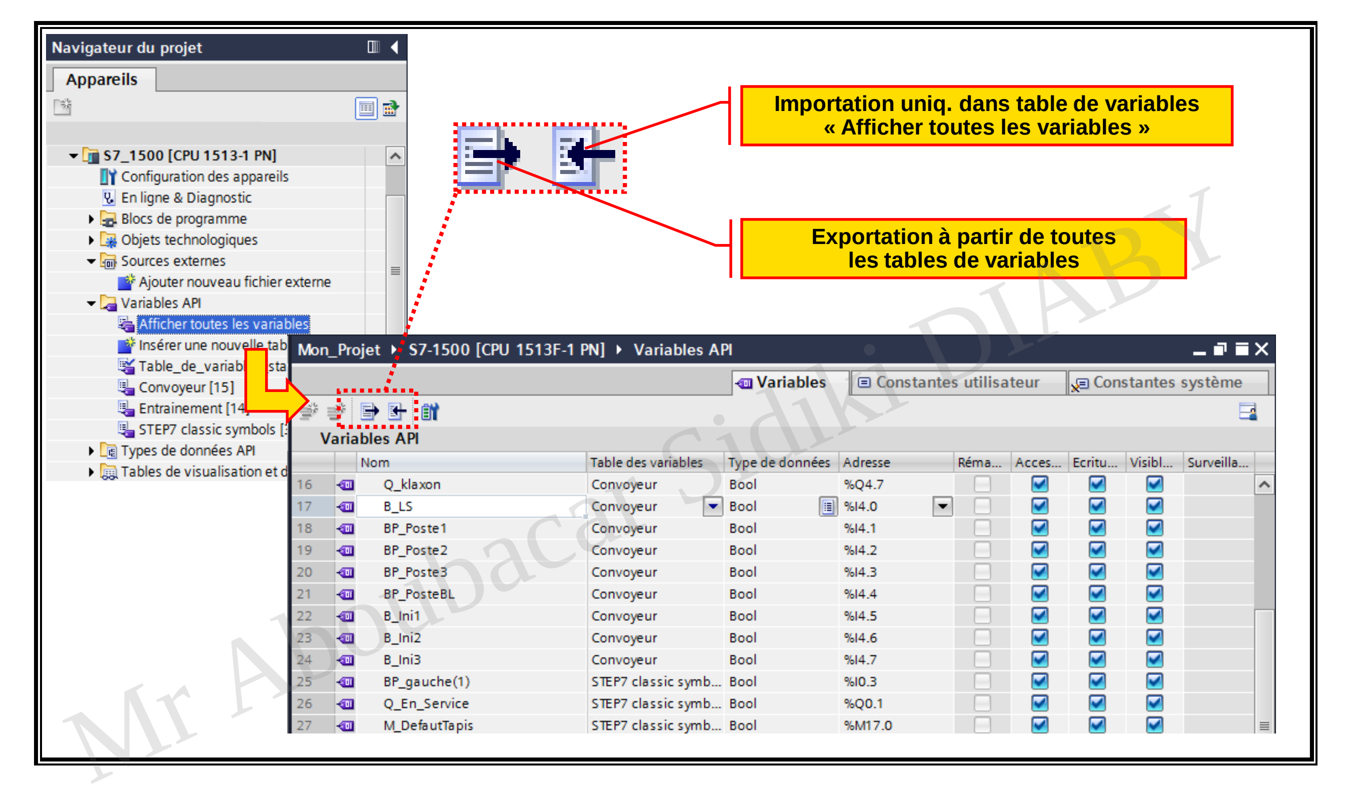
5.3.3. Signalisation d'erreurs dans une table de variables API



Contrôle syntaxique

Chaque saisie dans la table des variables fait l'objet d'un contrôle syntaxique. Les erreurs détectées sont signalées en ROUGE ou en JAUNE pour les avertissements. Une table de variables API contenant des erreurs peut être enregistrée, mais tant qu'elle n'a pas été corrigée, le programme ne peut être ni compilé ni chargé dans la CPU.

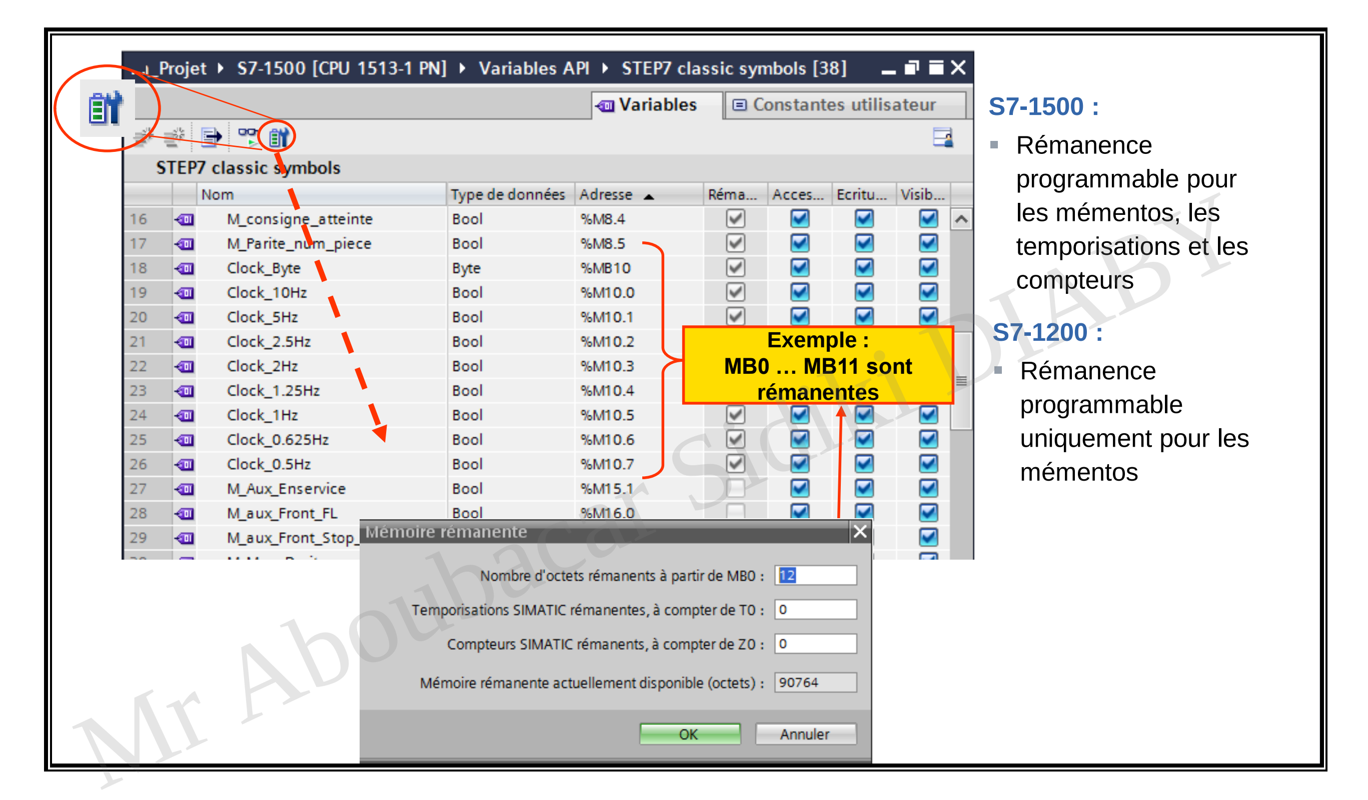
5.3.4. Copier & Coller des variables API vers Excel



Copier & Coller de et vers Excel

La fonction Copier & Coller de Windows peut être utilisée pour copier facilement de ou vers Excel une ou plusieurs variables sélectionnées dans une table de variables API ou un bloc de données.

5.3.5. Rémanence des variables API



Mémoire rémanente

Les CPU S7-1500 disposent d'une mémoire rémanente pour le stockage des données rémanentes en cas de coupure de la tension d'alimentation. La taille de cette mémoire rémanente est indiquée dans les caractéristiques techniques de la CPU.

L'occupation de la mémoire rémanente de la CPU configurée peut être visualisée hors ligne dans le navigateur du projet sous « Informations sur le programme > Ressources » ou en ligne dans le navigateur du projet sous « En ligne & Diagnostic > Diagnostic > Mémoire ».

Si vous définissez des données comme rémanentes, leur contenu sera conservé après une coupure de tension, lors du démarrage de la CPU et lors du chargement d'un programme modifié.

Les données ou objets suivants peuvent être définis comme rémanents :

Mémentos, temporisations, compteurs

Variables de blocs de données globaux

Variables de blocs de données d'instance d'un bloc fonctionnel

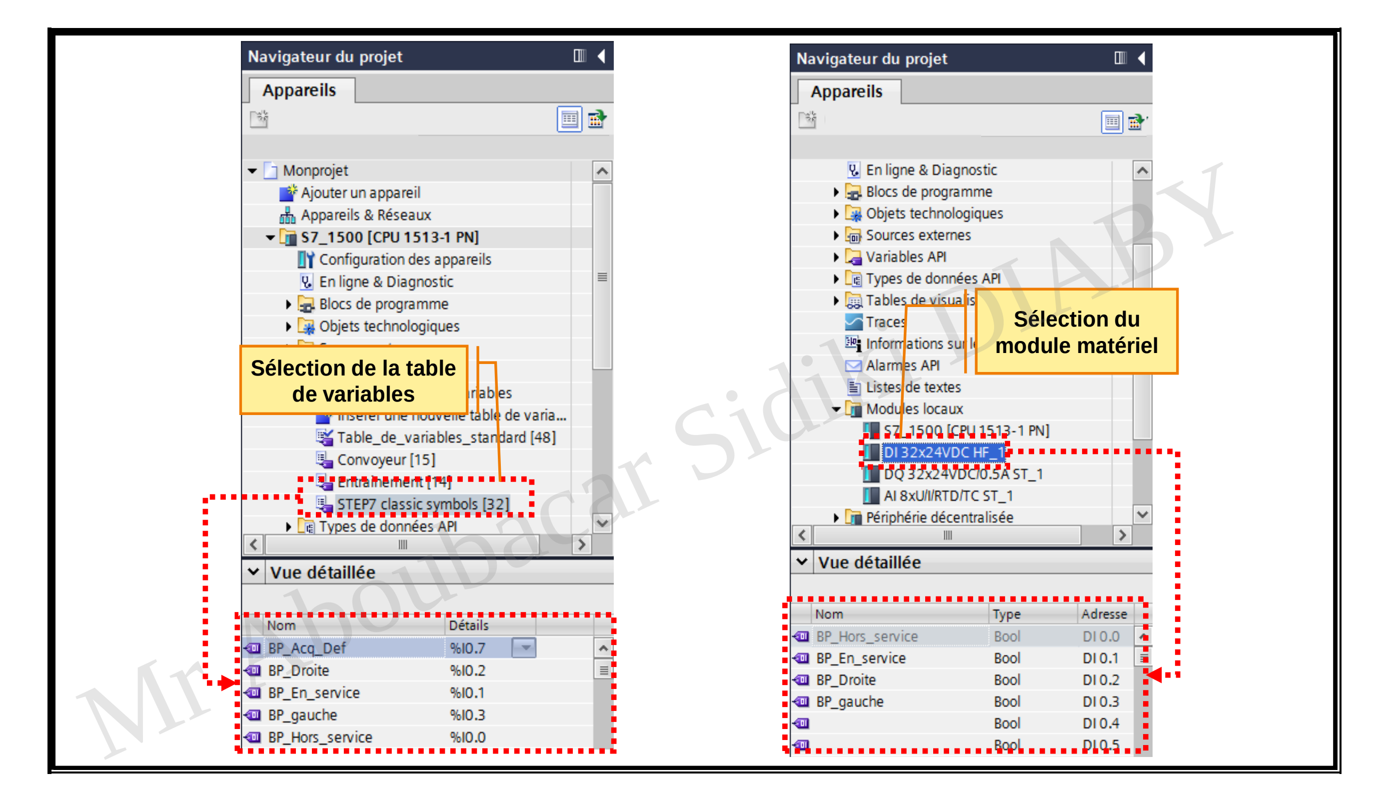
Certaines variables d'objets technologiques sont toujours rémanentes, comme les valeurs d'ajustage des codeurs absolus.

Mémentos, temporisations, compteurs

Avec le S7-1500, vous pouvez définir le nombre de mémentos, temporisations et compteurs rémanents dans la table de variables API via le bouton « Rémanence ».

Avec le S7-1200, vous ne pouvez définir que le nombre de mémentos rémanents dans la table de variables API via le bouton « Rémanence ».

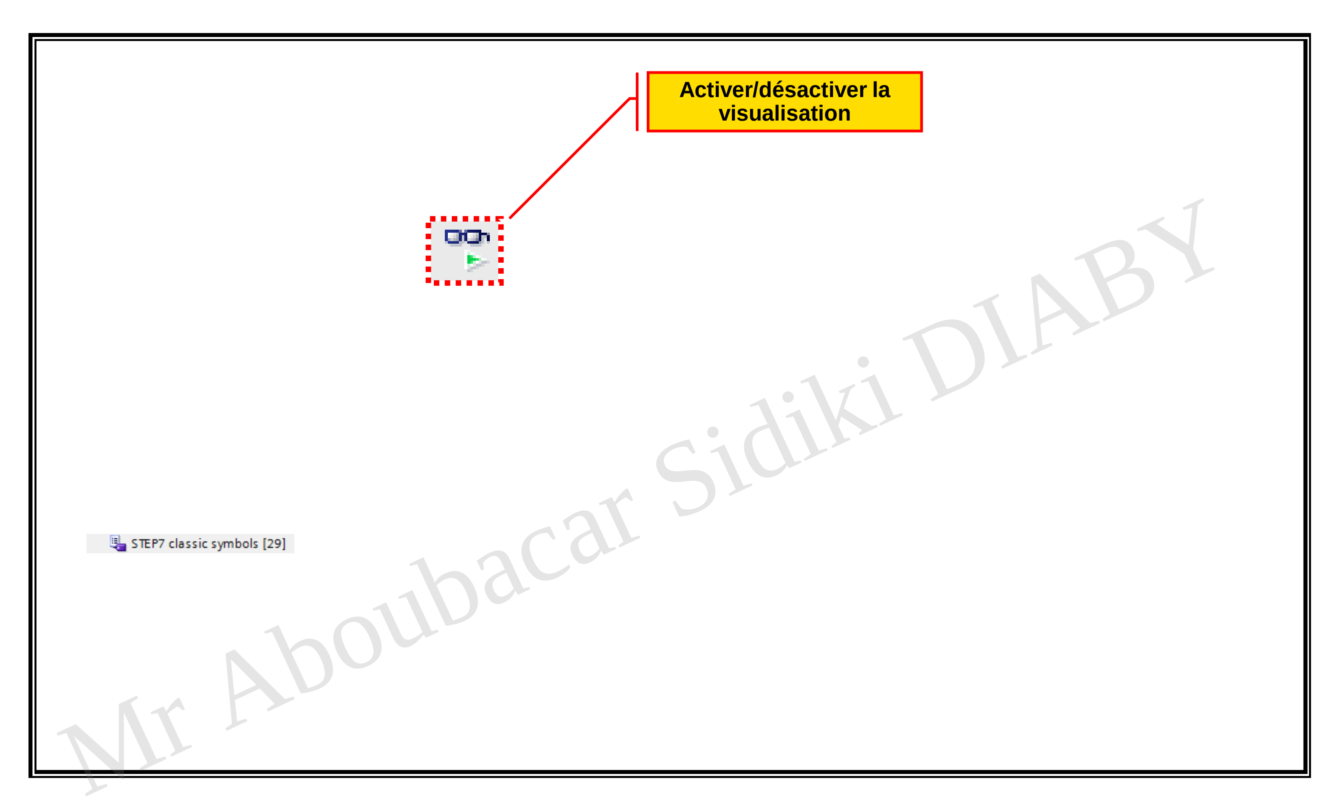
5.3.6. Vue détaillée des variables API



La vue détaillée permet de visualiser :

- les variables de la table de variables sélectionnée
- les variables du bloc de données sélectionné
- les voies des modules locaux sélectionnés et leurs variables

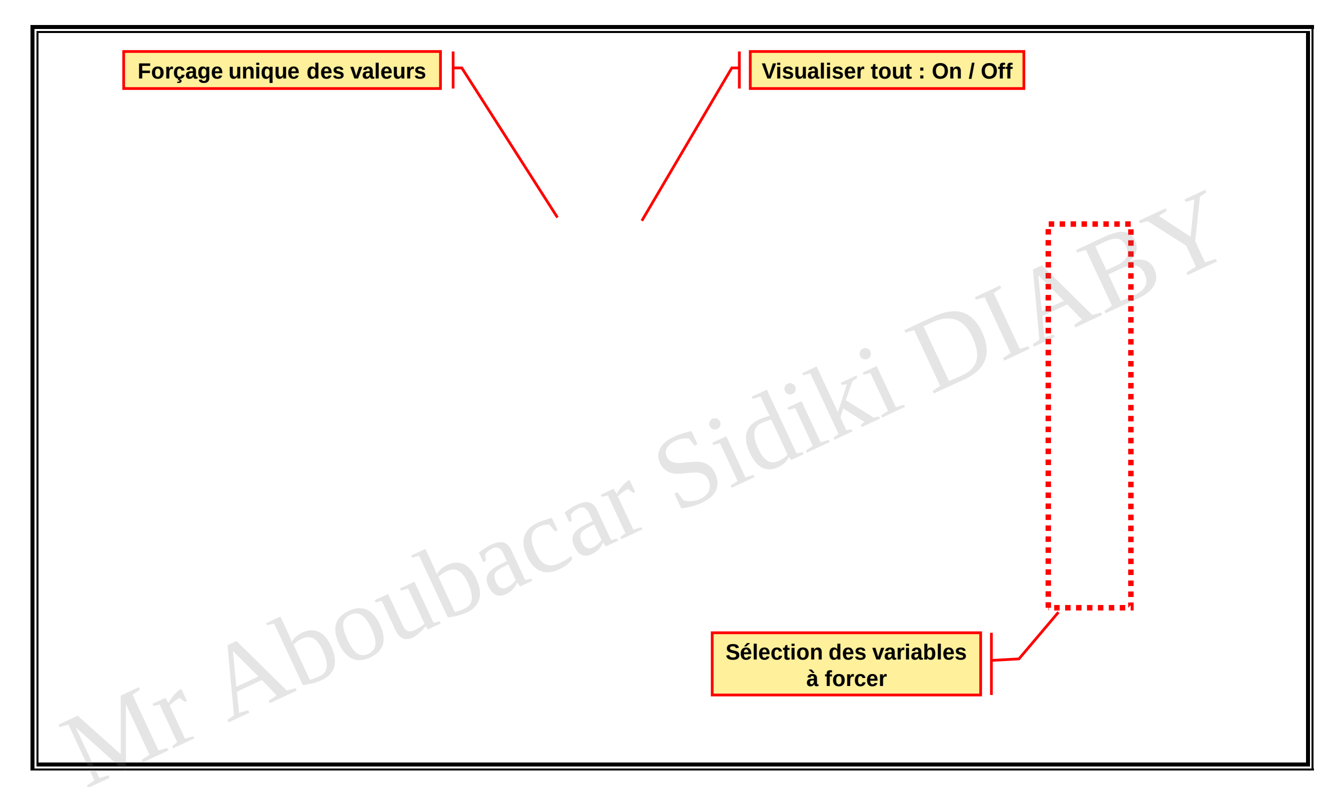
5.3.7. Visualiser des variables API



Visualiser des variables API

Les variables API peuvent être directement visualisées via la table de variables API. La « valeur de visualisation » indique la valeur courante des variables dans la CPU.

5.3.8. Modifier les variables API avec la table de visualisation



Pour modifier des variables il faut une table de visualisation que l'on créé dans « Table de visualisation ».

Toutes les variables peuvent être visualisées et pilotées dans la table. Pour modifier une variable, il faut spécifier une valeur de forçage, activer le pilotage de la variable (activation automatique lors de la création) et à l'aide du bouton « forcer les variables de manière unique et immédiate » les valeurs des variables sélectionnées sont chargées dans le contrôleur.

5.4. Exercice 1 : Copier des variables API depuis la bibliothèque globale



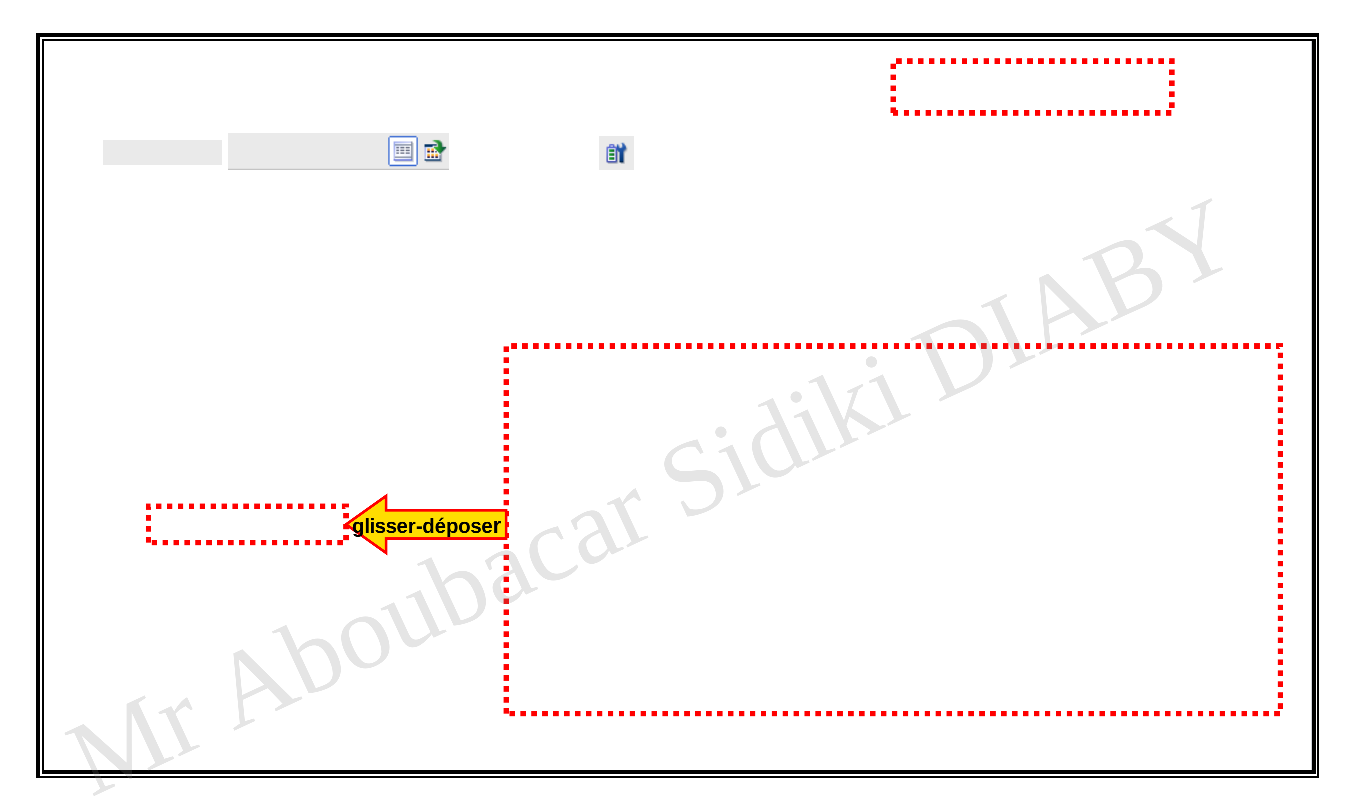
Enoncé

Il s’agit de copier dans le projet une table déjà préparée contenant des variables API depuis la bibliothèque globale « Bib_TIA_PRO1

Marche à suivre

1. Basculez vers la vue du projet.
2. Ouvrez la bibliothèque *<lecteur>:\Exercices_TIA_Portal_vxx\TIA_PRO1* dans l’onglet « Bibliothèques globales ».
3. Copiez par glisser&déposer l’objet « Mes_variables » contenu dans la bibliothèque globale « Bib_TIA_PRO1» vers votre projet dans le dossier « Variables API ».

5.4.1. Exercice 2 : créer une table de variables pour la maquette du convoyeur



Enoncé

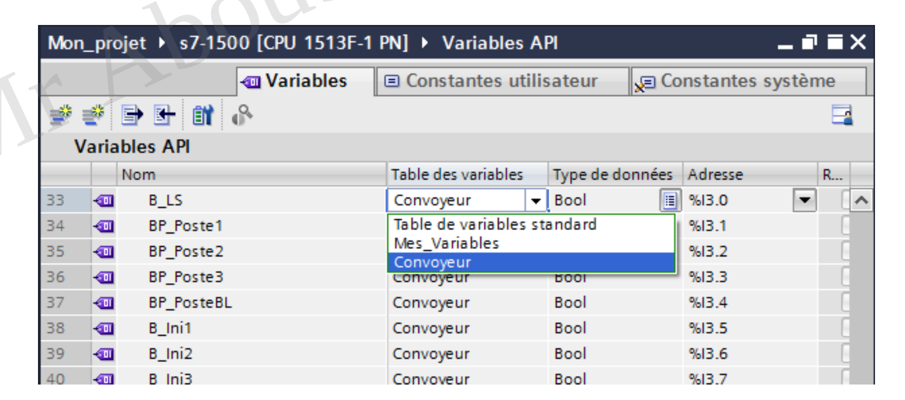
Une table de variables API séparée « Convoyeur » doit être créée pour les entrées et sorties auxquelles sont raccordés les capteurs et actionneurs de la maquette de bande.

Marche à suivre

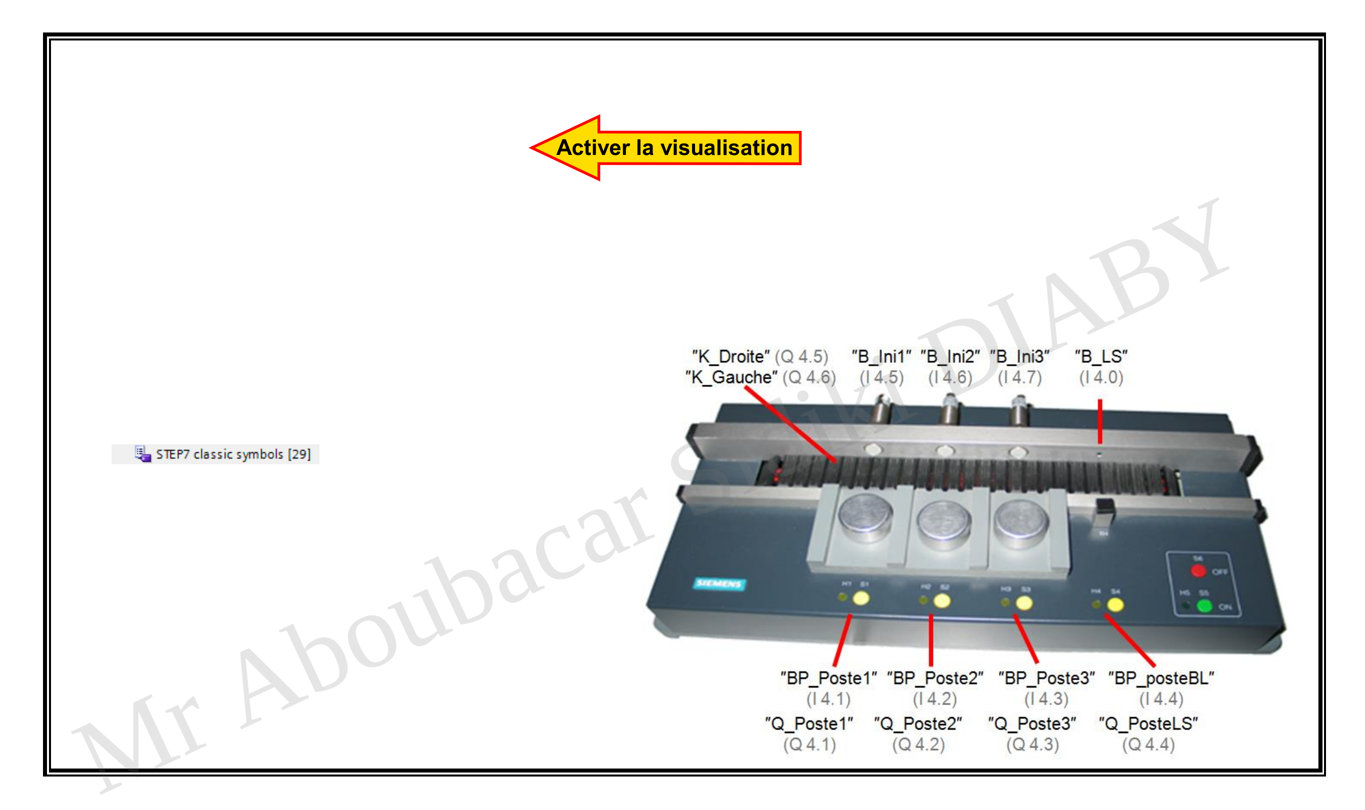
1. Créez la nouvelle table de variables API « Convoyeur » via « Insérer une nouvelle table de variables ».
2. Ouvrez la table de variables API « STEP7 classic symbols » et faites glisser avec la souris (pointeur de la souris sur les icônes de variables) les entrées/sorties dans la nouvelle table de variables API «Convoyeur » créée (voir figure).
3. Enregistrez votre projet.

Remarque:

Alternativement, vous pouvez affecter une table de variable spécifique à chaque variable. Cette affectation se réalise sous « Variables API », « Afficher toutes les variables» dans la colonne « Table des variables ».



5.4.2. Exercice 3 : visualiser la table de variables API « Convoyeur »



Enoncé

Veuillez visualiser les entrées dans la table de variables.

Marche à suivre

1. Afin de prendre en compte les modifications et pour que la CPU passe dans l'état RUN veuillez compiler et charger le programme.
2. Vérifiez que la connectique de votre convoyeur soit bien reliée au connecteur « S7-1500 DI/DO » au dos de votre poste de travail.
3. Veuillez mettre le convoyeur sous tension.
4. Ouvrez la table des variables API « Convoyeur » et activez la fonction « Visualiser tout ».
5. Au niveau du convoyeur manipulez les boutons poussoir des postes et vérifiez si l'état « 1 » ou « TRUE » est affiché aux entrées correspondantes.
6. Enregistrez votre projet.

5.4.3. Exercice 4: pilotage a partir de la table de visualisation



Enoncé

Veuillez piloter les sorties à l'aide d'une table de visualisation.

Marche à suivre

1. Au niveau de la table de visualisation et forçage ajoutez une nouvelle table de visualisation.
2. Ouvrez la table ainsi créée.
3. Dans le navigateur du projet sélectionnez la table de variables API « convoyeur ».
4. Copiez les variables API de « Convoyeur » par glisser – déposer dans votre table de visualisation.
5. Pour quelques sorties spécifiez la valeur de forçage TRUE ou 1.
6. Vérifiez si les variables sont validées pour le forçage.
7. Activez la fonction « Visualiser tout » et modifiez les sorties.
8. Vérifiez si les entrées correspondantes sont activées.
9. Pour les sorties passez les valeurs de forçage à FALSE ou 0 pour les désactiver
10. Modifiez encore une fois les sorties.
11. Enregistrez votre projet.

Sommaire 6

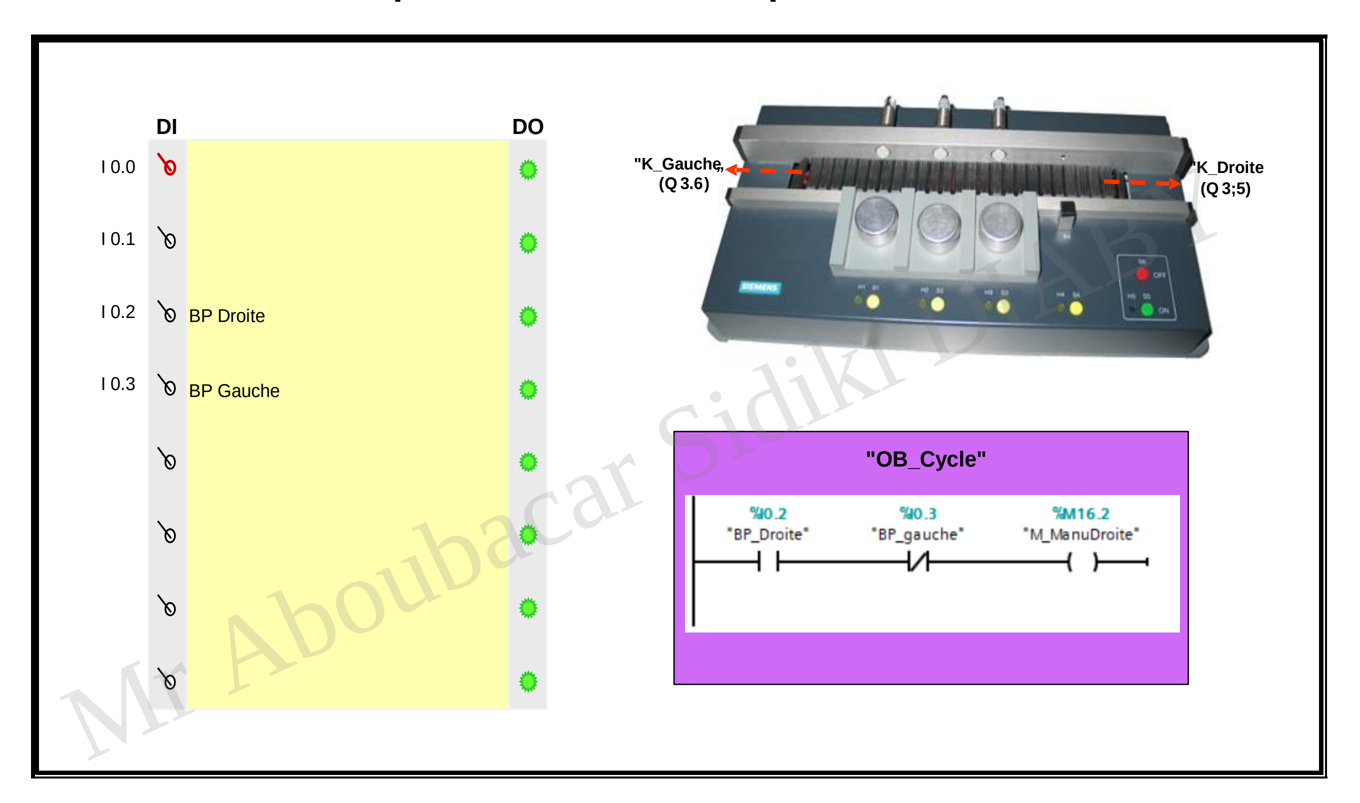
6. Opérations binaires

A l'issue du chapitre, vous allez ...

…	savoir ce que représente une assignation
...	Comprendre la différence entre des contacts NF et NO câblés et les instructions de traitement utilisées
…	connaître et utiliser les opérateurs logiques ET , OU et OU exclusif
…	connaître la mémoire image du processus pour les entrées et les sorties
...	comprendre l'exécution cyclique d'un programme
…	connaître la signification de « programme linéaire »
…	savoir utiliser l'éditeur de programme
…	être en mesure de renommer et de réaffecter les variables API
...	pouvoir tester un programme avec la fonction « Visualisation du programme »



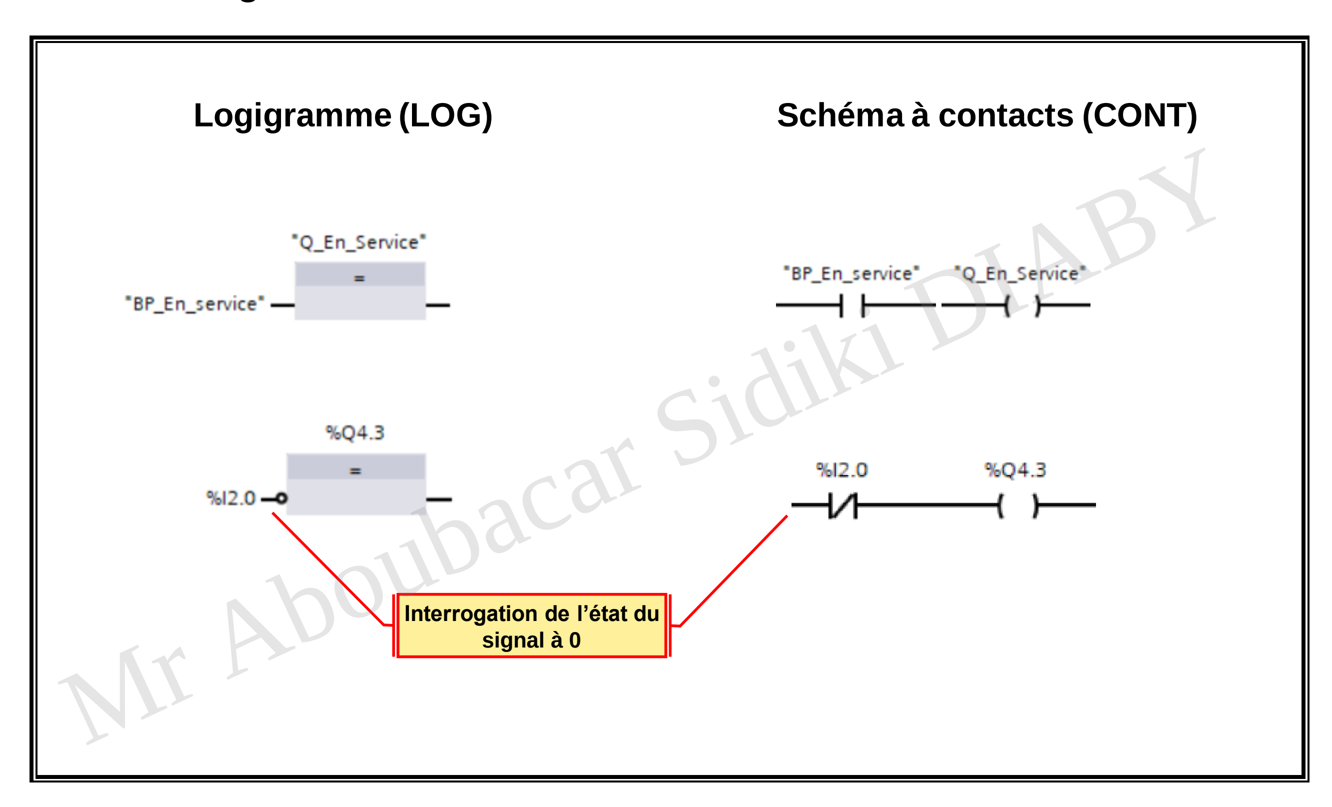
6.1. Enoncé : Maquette de bande transporteuse



Maquette de bande transporteuse

Le moteur de la bande peut être commandé par impulsions vers la droite et vers la gauche via les boutons du simulateur « BP_droite » (I0.2) et « BP_gauche » (I0.3) Dans le programme un verrouillage doit être mis en place pour éviter que la bande bouge lorsque les 2 boutons sont appuyés simultanément.

6.2. Assignation



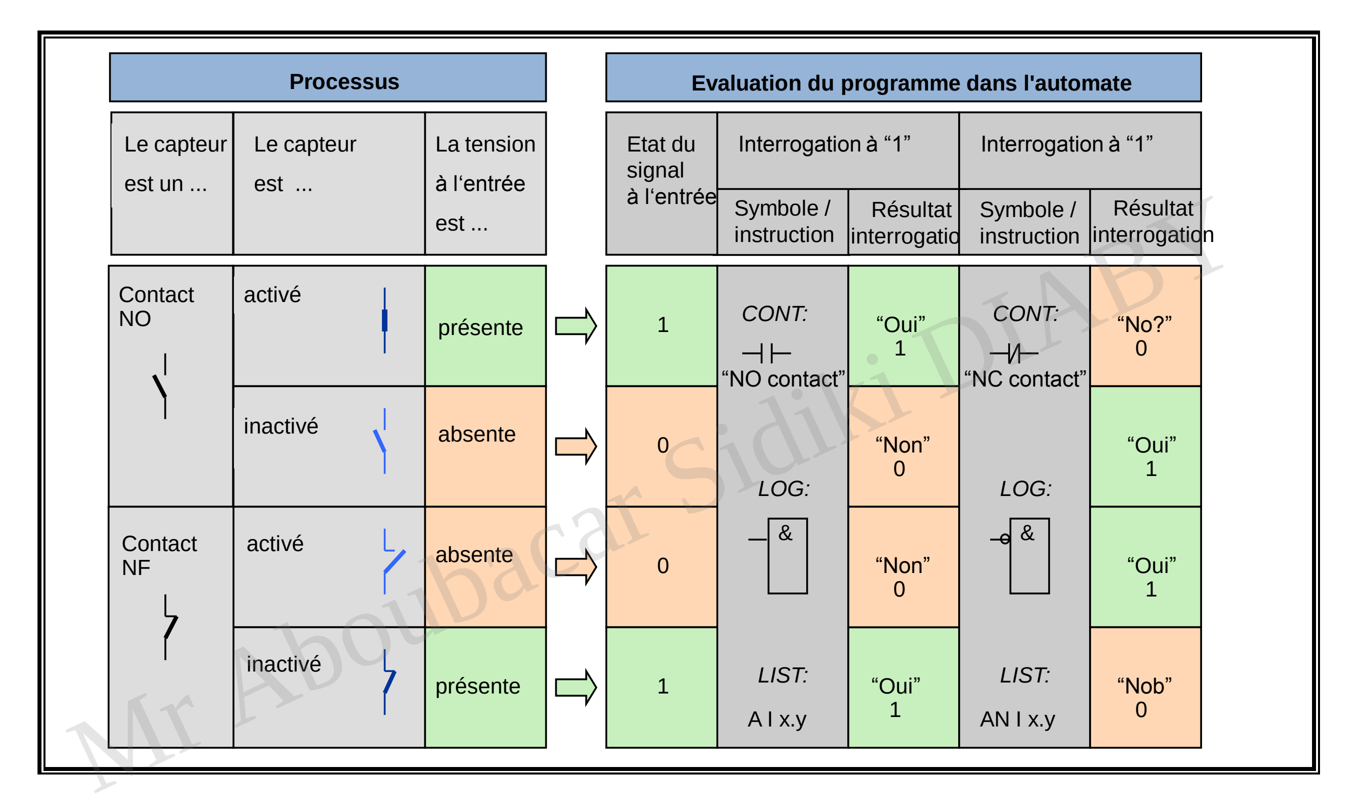
Assignation

Avec l'assignation, l'opérande spécifié prend l'état du résultat logique en cours. Le résultat logique reste disponible après l'assignation et peut être assigné à un autre opérande ou peut être combiné logiquement ultérieurement.

Interrogation à 0 (False)

L'interrogation à 0, c'est-à-dire, l'état du signal « FALSE » est satisfait lorsque l'opérande présente un état de signal « 0 ».

6.2.1. Capteurs et instructions d’interrogation



Capteurs utilisés par le processus

Le recours à des contacts NF ou NO pour les capteurs est fonction des règles de sécurité applicables au processus. Par principe, les fins de course et les interrupteurs de sécurité utilisent des capteurs à contact NF (normalement fermé) pour prévenir les risques en cas de rupture de câble. C’est aussi la raison pour laquelle on utilise des contacts NF pour la mise hors tension des installations.

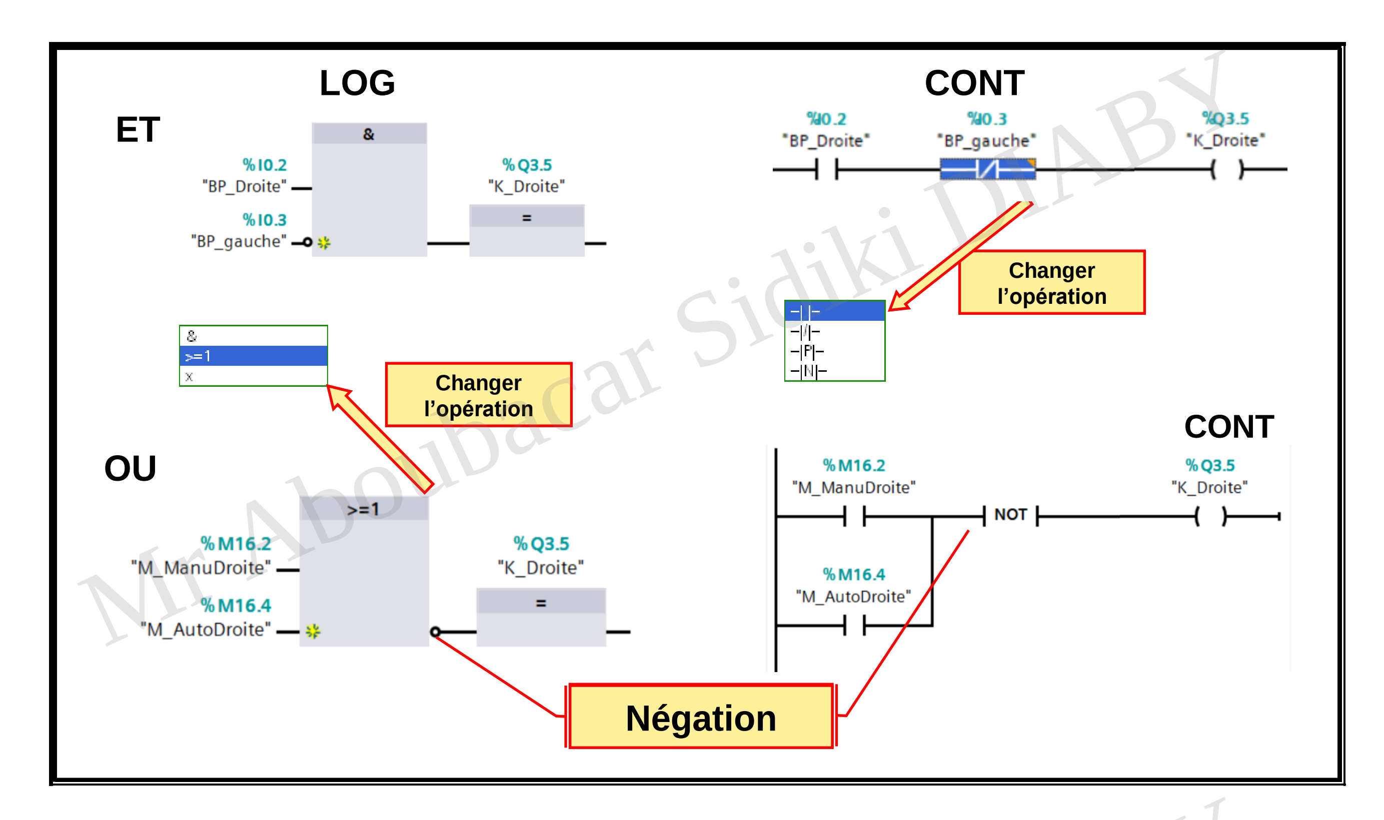
Instructions d’interrogation

En CONT, on utilise un symbole de « contact NO » pour l‘interrogation du signal à « 1 » et un symbole de « contact NF » pour l‘interrogation du signal à « 0 ». Il importe peu de savoir si le signal « 1 » du processus est délivré par un contact NO activé ou un contact NF inactivé.

Le symbole d’interrogation « contact NO » fournit le résultat d’interrogation ou le résultat logique (RLG) « 1 » lorsque l’opérande interrogé est à l’état « 1 ».

Le symbole d'interrogation « contact NF » fournit le résultat d’interrogation ou le résultat logique (RLG) « 1 » lorsque l'opérande interrogé est à l'état « 0 ».

6.3. Opérations logiques binaires : ET, OU, et NEGATION



Opérations ET et OU

Les opérations ET et OU permettent en principe d'interroger tous les opérandes binaires, y compris les sorties. Elles peuvent également relier les résultats d'autres opérations et sont combinables.

Toutes les entrées des opérations peuvent être programmées pour interroger un état de signal ou un état à « 0 » ou « 1 », et ce indépendamment de la nature du contact matériel raccordé au niveau du processus (NO ou NF).

Opération ET

Dans une opération ET, le résultat logique (RLG) est égal à « 1 » lorsque **tous** les signaux d'entrée sont à l'état « 1 ».

Opération OU

Dans une opération OU, le résultat logique (RLG) est égal à « 1 » lorsqu'**au moins un** signal d'entrée est à l'état « 1 ».

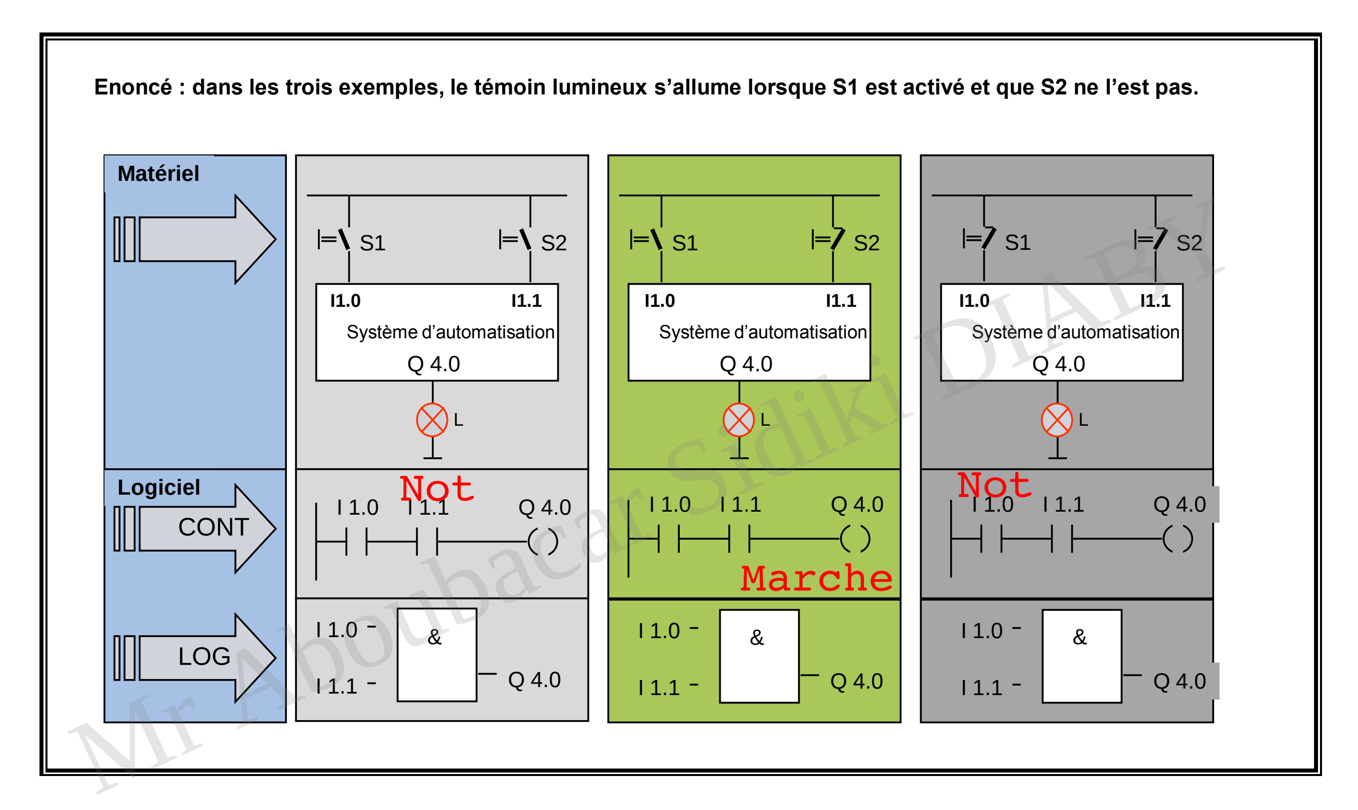
NOT

L'opération NOT inverse le RLG.

Dans l'exemple ci-dessus, si le RLG de l'opération ET est égal à « 1 », l'opération NOT l'inverse à « 0 » et l'opération de mise à 1 n'est pas exécutée (l'état de « K_Droite » (Q3.5) reste alors inchangé).

Si le RLG de l'opération ET est égal à « 0 », l'opération NOT l'inverse à « 1 » et l'opération de mise à 1 est exécutée (« K_Droite » (Q3.5) est mis à « 1 »).

6.3.1. Exercice théorique 1 : Symboles de capteurs et d'interrogations



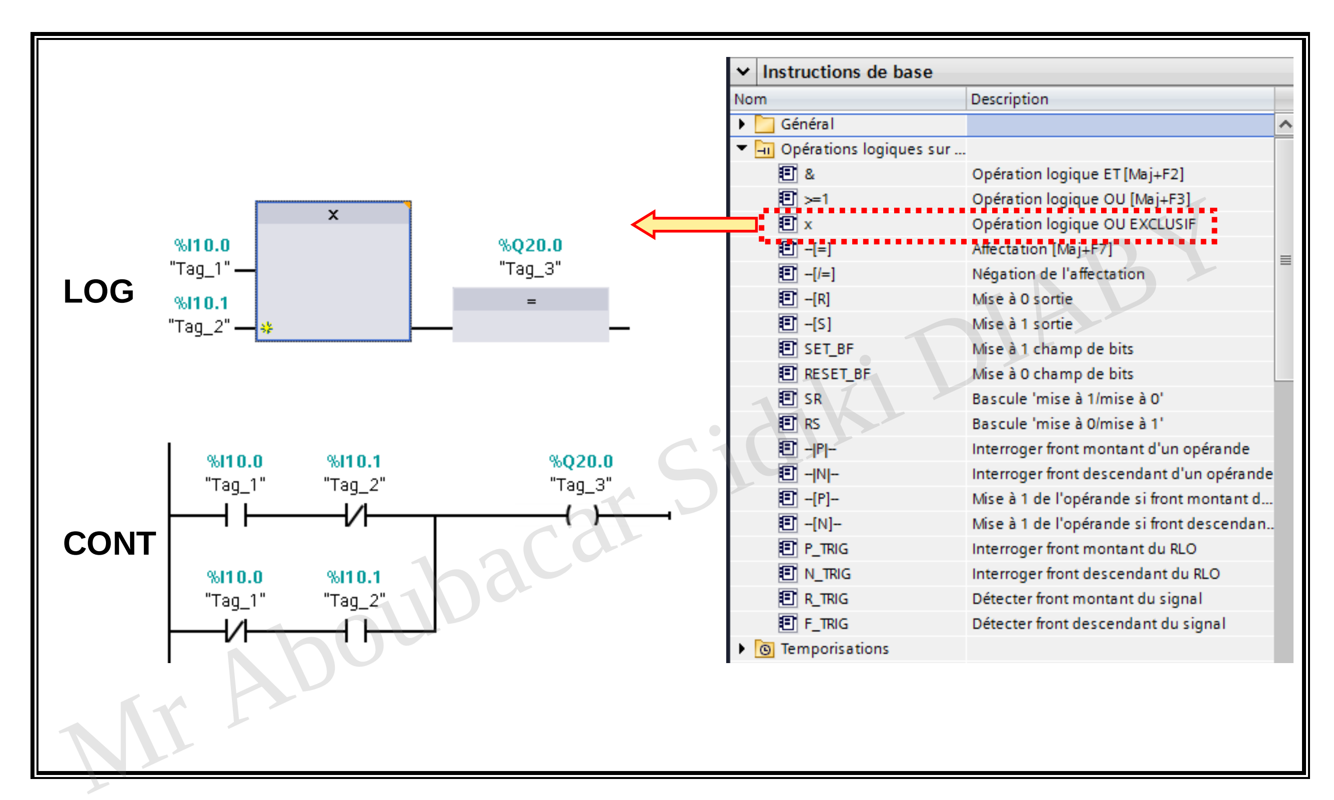
Enoncé

Complétez les symboles d'opération en LOG et corrigez les symboles d'interrogation en CONT pour que la fonction indiquée ci-dessus soit assurée.

Remarque

Les termes « contact NF » et « contact NO » ont une signification différente selon qu'il s'agit de contacts employés dans l'environnement matériel du processus ou d'instructions d'interrogation dans le logiciel.

6.3.2. Opérations logiques binaires : OU exclusif (XOR)



Opération XOR

L'opération XOR permet en principe d'interroger tous les opérandes binaires, y compris les sorties. Elle peut également relier les résultats d'autres opérations à la place des opérandes et est combinable.

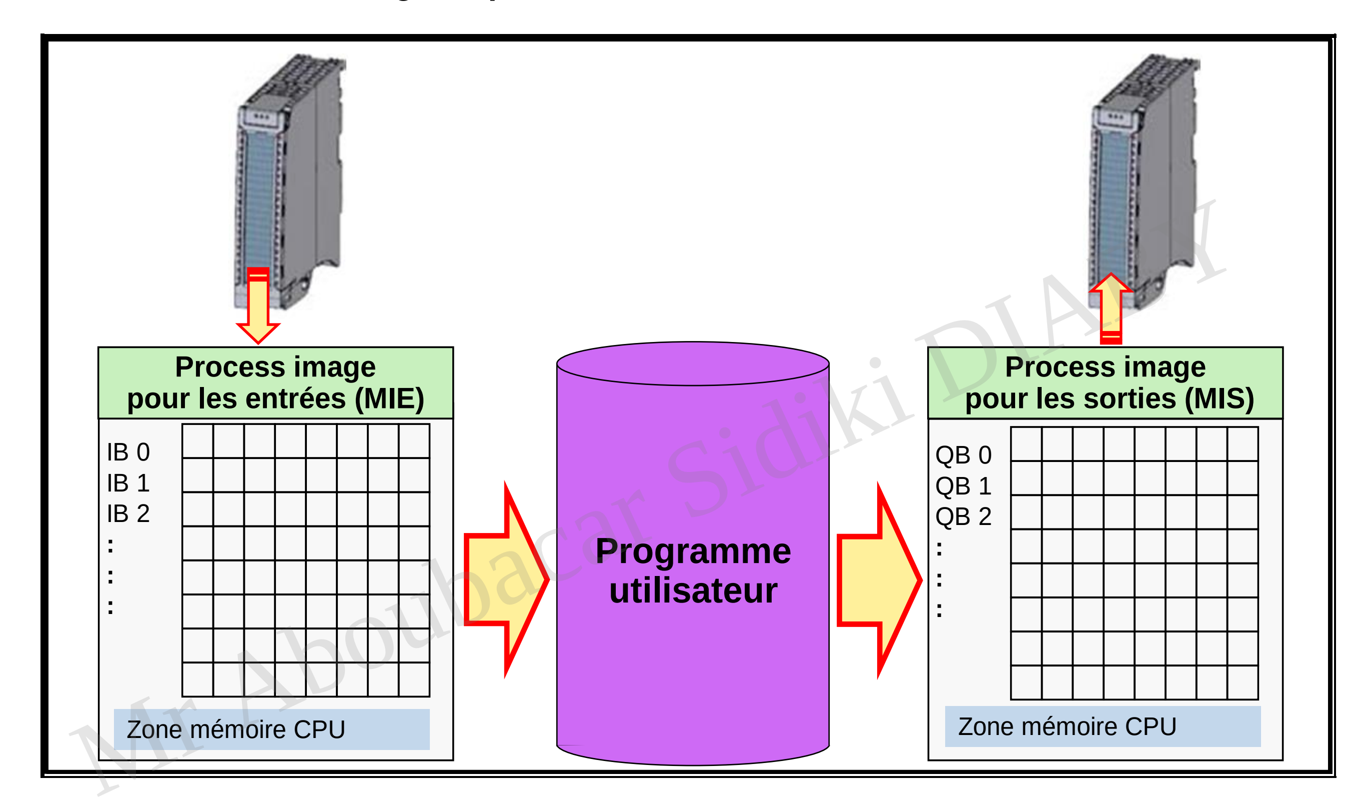
Toutes les entrées de l'opération peuvent être programmées pour interroger un état de signal ou un état à « 0 » ou « 1 », et ce indépendamment de la nature du contact matériel raccordé au niveau du processus (NO ou NF).

- En cas d'opération XOR entre 2 opérandes, le résultat logique (RLG) est égal à « 1 » lorsqu'un et un seul des deux opérandes est à « 1 ».
- En cas d'opération XOR entre plus de 2 opérandes, le RLG est...
  - égal à « 1 » si un nombre impair d'opérandes interrogés est à « 1 »
  - égal à « 0 » si un nombre pair d'opérandes interrogés est à « 1 ».

XOR en LOG et CONT

Le langage de programmation CONT ne dispose d'aucune opération XOR explicite. Celle-ci doit être programmée à l'aide d'instructions discrètes, comme indiqué sur la figure ci-dessus.

6.3.3. Mémoires image du processus



Mémoires image du processus

Pour stocker les états logiques des entrées et sorties TOR interrogées, la CPU dispose de zones mémoire réservées : la mémoire image des entrées (MIE) et la mémoire image des sorties (MIS). Lors du traitement, le programme utilisateur accède à ces zones mémoire, et non directement aux modules d'entrées et sorties TOR.

Mémoire image des entrées (MIE)

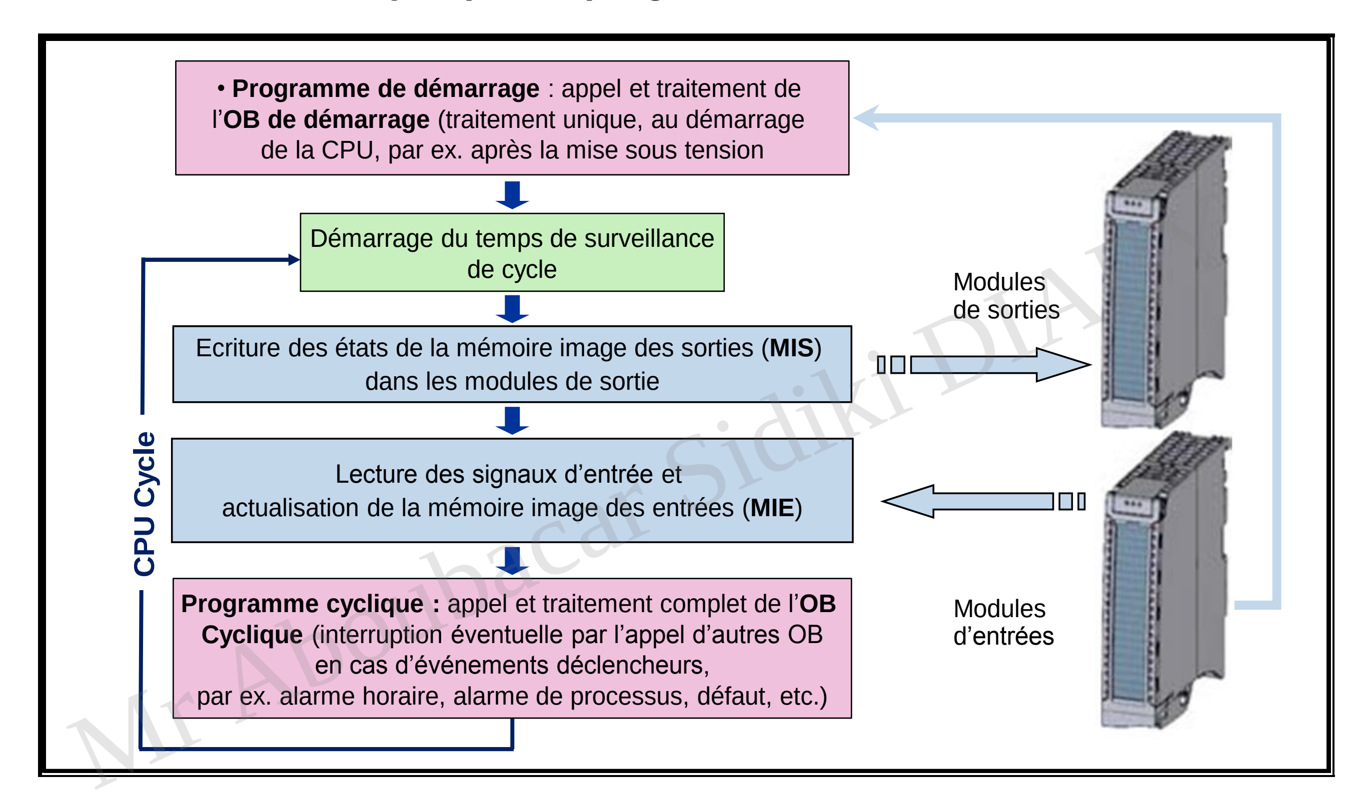
La mémoire image des entrées (MIE) est la zone mémoire dans laquelle sont stockés les états de toutes les entrées. Ces états sont lus en début de cycle sur les modules d'entrées et transmis à la MIE. En cas d'opération sur une entrée, le programme utilisateur utilise l'état de l'entrée stocké dans la MIE. Cet état ne peut varier en cours de cycle, car la MIE n'est actualisée et lue qu'en début de cycle. Ainsi, si une entrée est interrogée plusieurs fois au cours d'un même cycle, elle délivrera toujours le même résultat.

Mémoire image des sorties (MIS)

La mémoire image des sorties (MIS) est la zone mémoire dans laquelle sont stockés les états de toutes les sorties. En fin de cycle, la MIS est transmise aux modules aux modules de sortie.

Dans le programme, les sorties peuvent être aussi bien affectées (écrites) qu'interrogées (lues).

6.4. Traitement cyclique du programme



Redémarrage

A la mise sous tension ou en cas de commutation de STOP --> RUN, la CPU procède à un redémarrage complet pendant lequel sont exécutés les OB de démarrage programmés. Lors du redémarrage, le système d'exploitation efface les mémentos non rémanents et après l'exécution des OB de démarrage il lance la surveillance du temps de cycle. **(Les OB de démarrage sont traités dans le chapitre « Blocs d'organisation »)**

Traitement cyclique du programme

Le traitement cyclique du programme s'exécute dans une boucle sans fin. Après la fin du traitement d'un cycle de programme, le traitement du cycle suivant est lancé automatiquement. A chaque cycle de traitement du programme, la CPU traite les étapes suivantes.

- Transfert des états des sorties de la mémoire image des sorties vers les modules de sorties TOR.
- Lecture des états des entrées et actualisation de la mémoire image des entrées.
- Exécution des instructions du programme utilisateur. Ce traitement a lieu principalement avec les mémoires image des E/S, et non directement avec les modules d'entrées et de sorties.

Temps de cycle et temps de surveillance de cycle

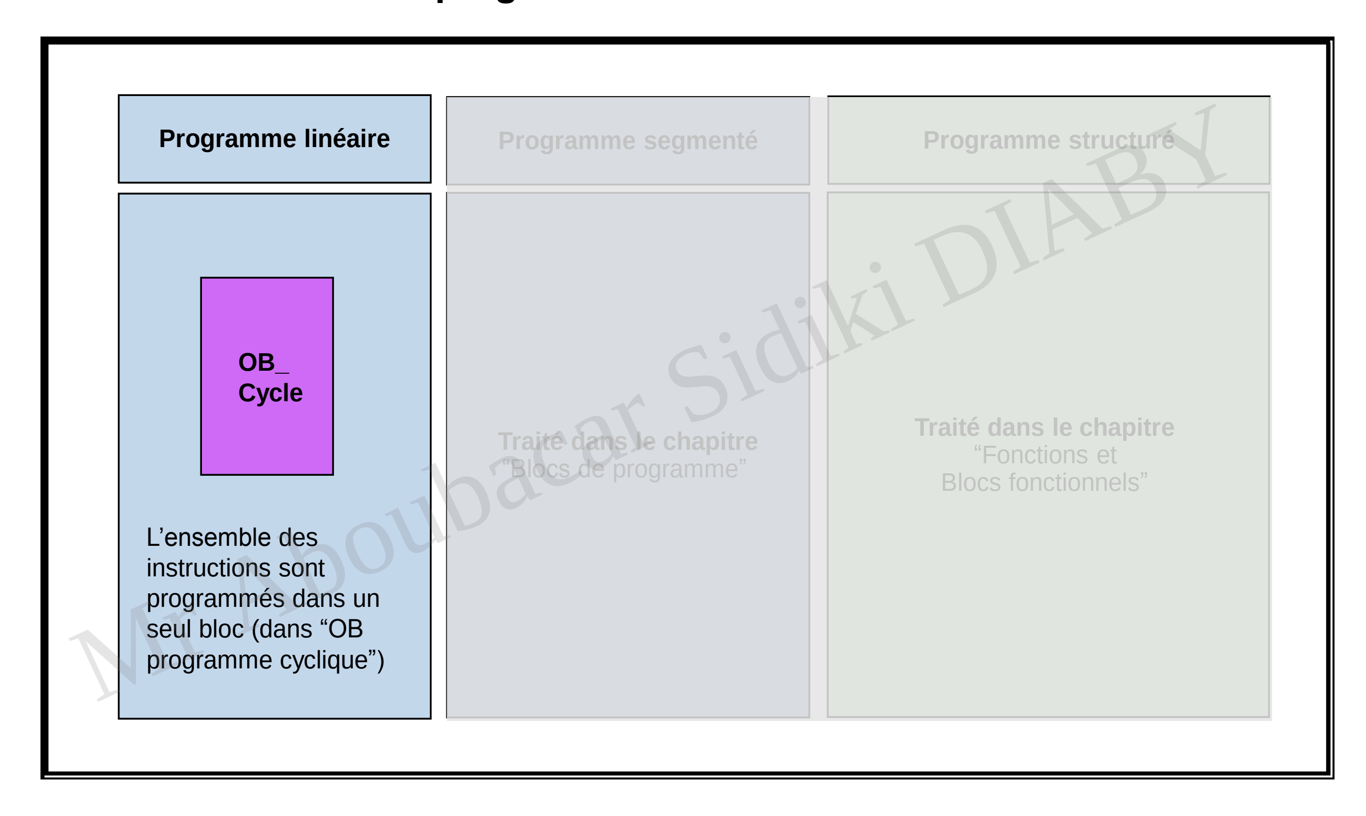
Le temps nécessaire à la CPU pour exécuter un cycle complet du programme est appelé temps de cycle. Il est surveillé par le système d'exploitation de la CPU. Si le temps de cycle dépasse le temps de surveillance défini dans les propriétés de la CPU **(Chapitre Appareils et Réseaux),** un « OB d'interruption d'erreur de temps de surveillance » est appelé. Si cet OB n'est pas configuré ou si le temps est le double du temps programmé alors la CPU passe à l'état STOP.

Utiliser la mémoire image ou la désélectionner pour des modules spécifiques

Dans les propriétés des modules vous pouvez sélectionner si les valeurs des entrées doivent être automatiquement transférées dans la mémoire image des entrées ou encore si les sorties doivent

être automatiquement actualisées avec les valeurs de la mémoire image des sorties (au début du cycle), et cela uniquement si un OB spécifié est traité ou jamais.

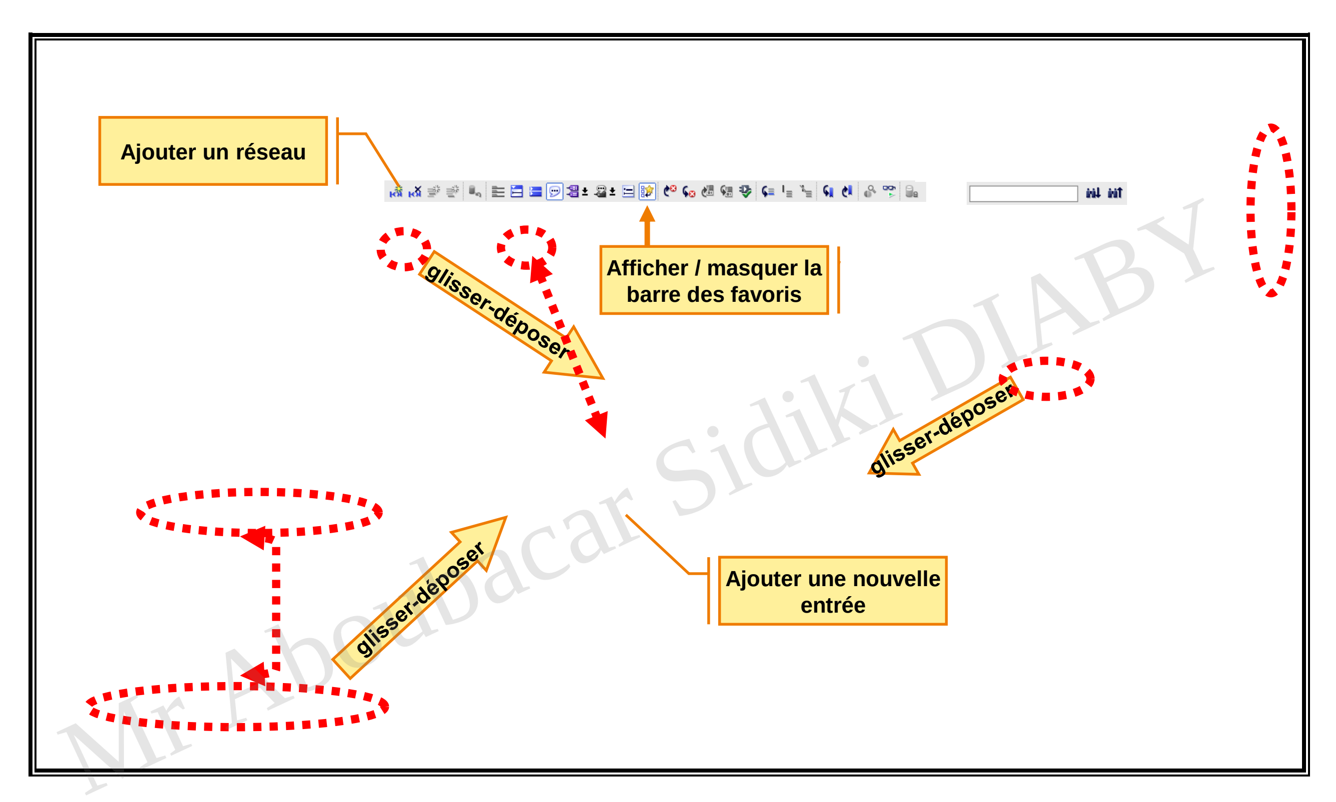
6.5. Structure d'un programme



Programme linéaire

L'ensemble des instructions du programme se trouve dans un seul bloc. La CPU traite successivement les différentes instructions. Ce type de programmation s'apparente à la logique câblée.

6.5.1. Programmation des blocs : instructions



Instructions

Les instructions d'un bloc peuvent être programmées comme suit :

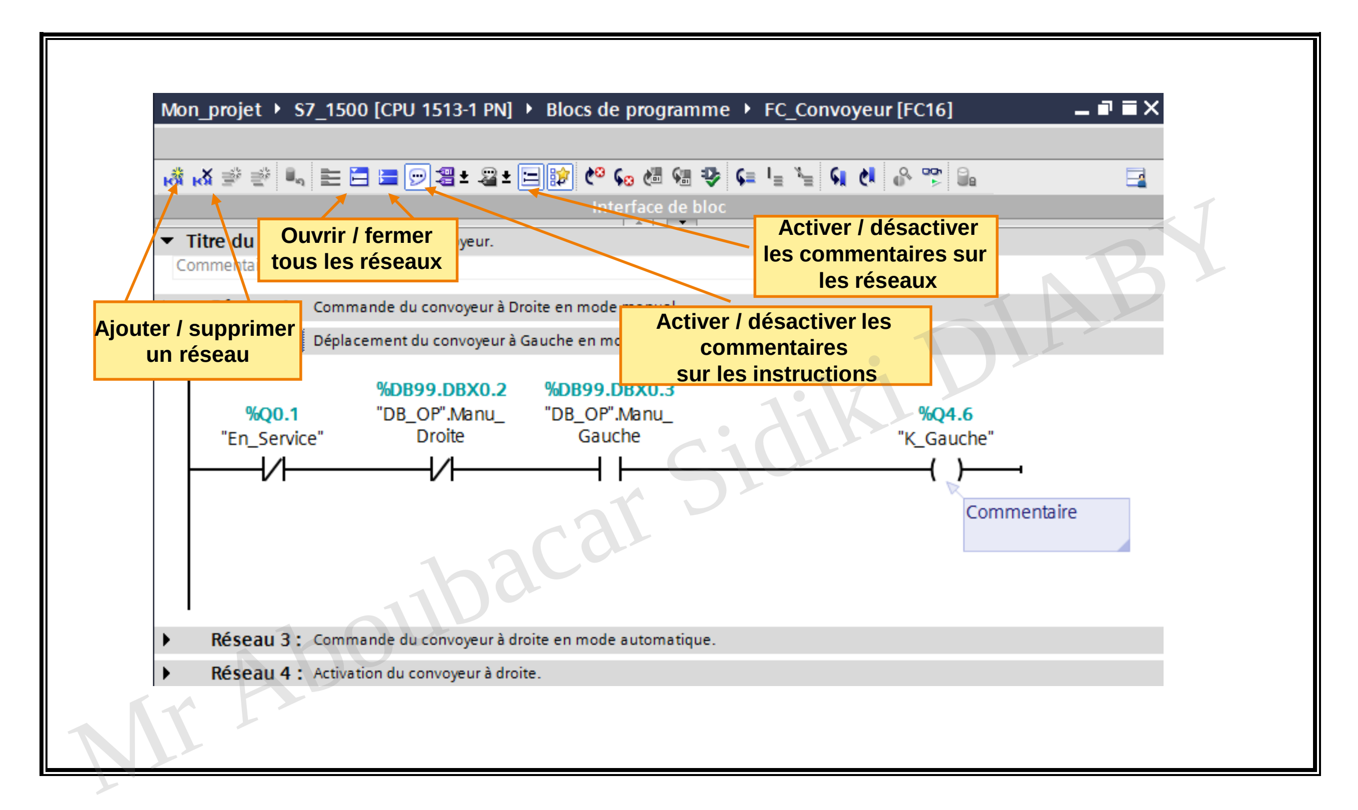
- par glisser-déposer en un point quelconque du programme à partir de la barre des favoris ou du catalogue des instructions (Task Card « Instructions »)
- par sélection de l'emplacement dans le programme, puis double-clic sur l'instruction souhaitée dans la barre des favoris ou le catalogue des instructions (Task Card « Instructions »)

Les opérandes peuvent être entrés avec leur adresse absolue ou symbolique. Si la table des variables est sélectionnée (pas ouverte !) dans le navigateur du projet, des variables peuvent être insérées dans le programme à l'emplacement souhaité par glisser-déposer à partir de la « Vue détaillée ».

Barre des favoris

Les éléments LOG fréquemment utilisés sont disponibles dans la barre des favoris. Celle-ci peut être étendue par glisser-déposer à partir du catalogue des instructions.

6.5.2. Programmation des blocs : réseaux

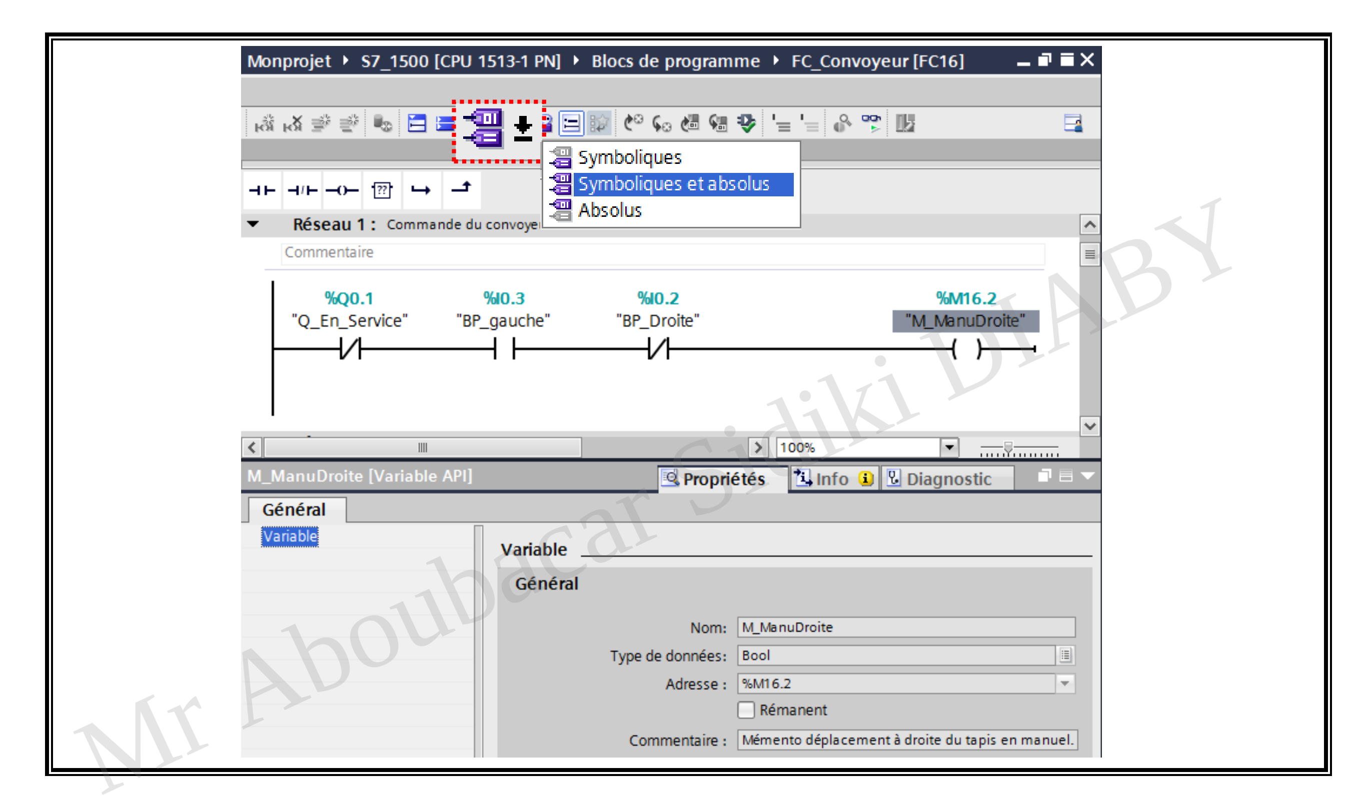


Réseaux

Tout comme le programme utilisateur peut être subdivisé en blocs, chaque bloc peut à son tour être subdivisé en différents réseaux. Cette subdivision en réseaux est librement déterminée par l'utilisateur.

Chaque réseau peut être doté d'un titre de réseau et d'un commentaire de réseau. Au sein de ces réseaux, les différentes instructions peuvent être dotées de commentaires d'instruction.

6.5.3. Adressage absolu et symbolique



Adressage absolu et symbolique

Toutes les variables globales (par ex. entrées, sorties, mémentos) possèdent une adresse absolue et une adresse symbolique. L'utilisateur peut déterminer l'adresse qu'il veut visualiser à l'écran et le type d'adressage qu'il souhaite utiliser pour la programmation (voir figure ci-dessus).

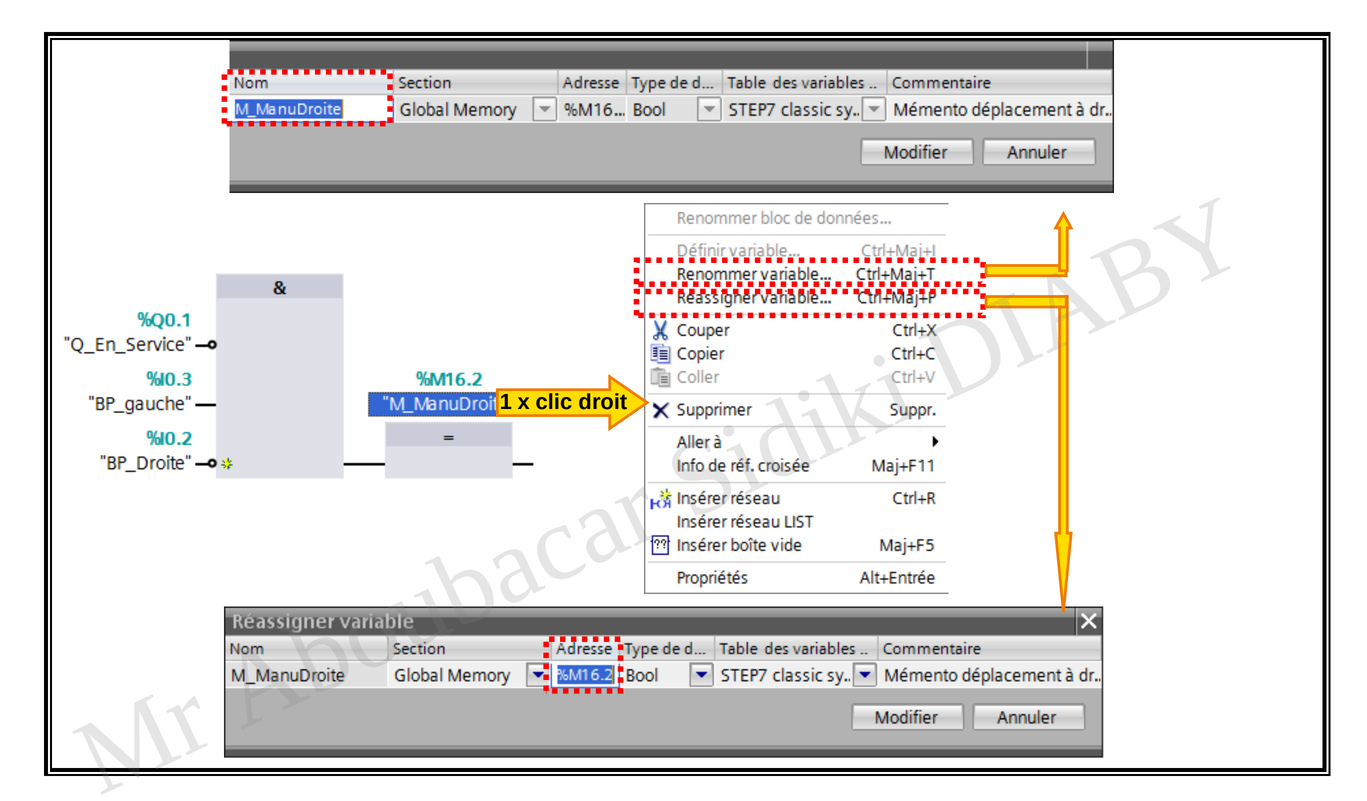
Si une adresse symbolique (par ex. « M_ManuDroite ») est utilisée dans le programme sans qu'une adresse absolue ne lui ait été affectée, le bloc peut être enregistré, mais la compilation et le chargement dans l'automate sont impossibles.

Si une adresse absolue est utilisée dans le programme (par ex. M16.2), le système lui affecte automatiquement une adresse symbolique standard (par ex. « Var_1 »). Celle-ci peut être modifiée ultérieurement.

Propriétés

Si l'on sélectionne une variable dans un bloc ou une table de variables API ouverts dans la fenêtre de travail, tous les détails éditables relatifs à cette variable s'affichent dans la fenêtre d'inspection sous l'onglet « Propriétés ».

6.5.4. Renommer / réassigner des variables API

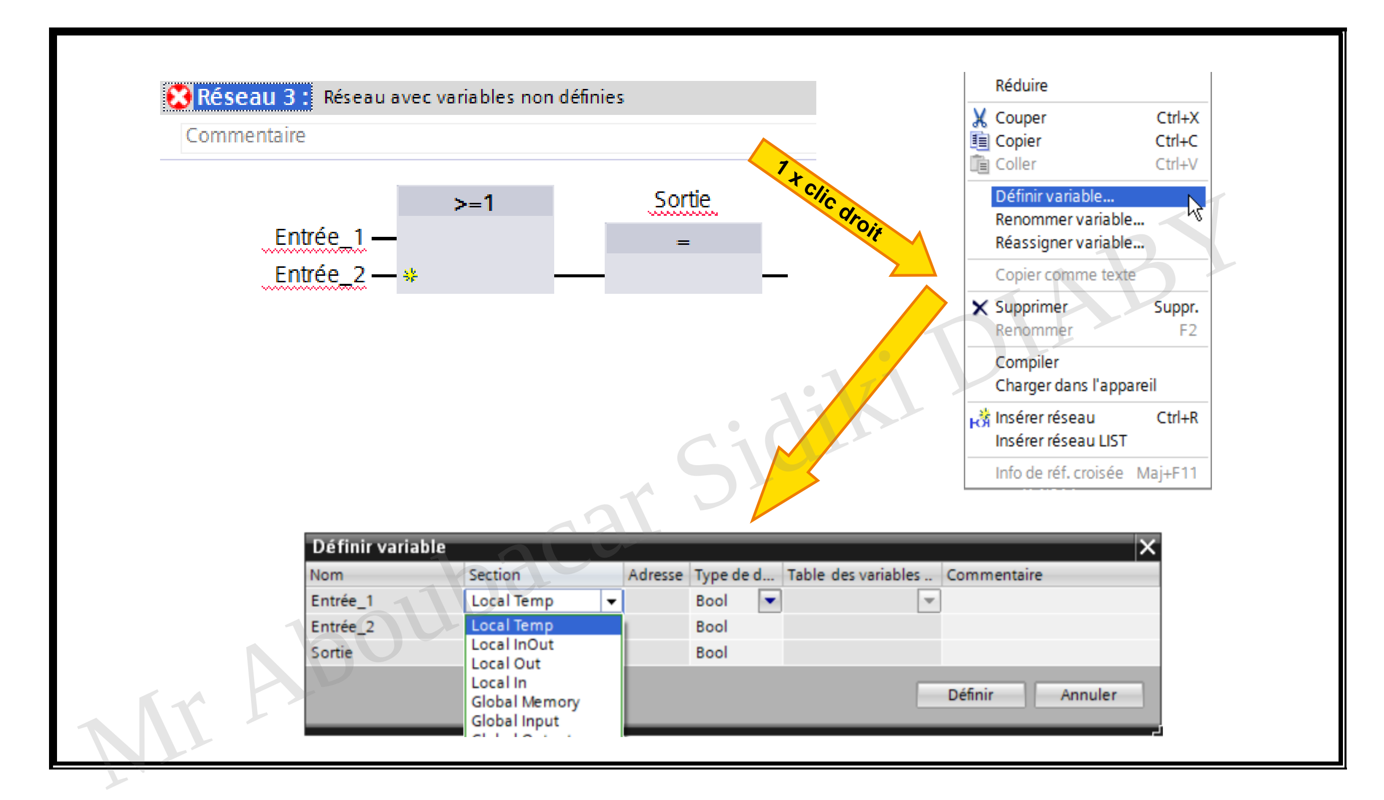


Renommer et réassigner des variables

Les variables peuvent être directement renommées ou réassignées dans la table de variables API ou, comme indiqué dans la figure ci-dessus, via l'éditeur de blocs. Les modifications sont immédiatement prises en compte dans la table de variables API et sont appliquées à l'ensemble du programme.

- **Renommer** :
  Modification du nom de la variable sans changement de l'adresse absolue
- **Réassigner**:
  Modification de l'adresse absolue affectée à une variable sans changement du nom

6.5.5. Déclarer des variables lors de la programmation



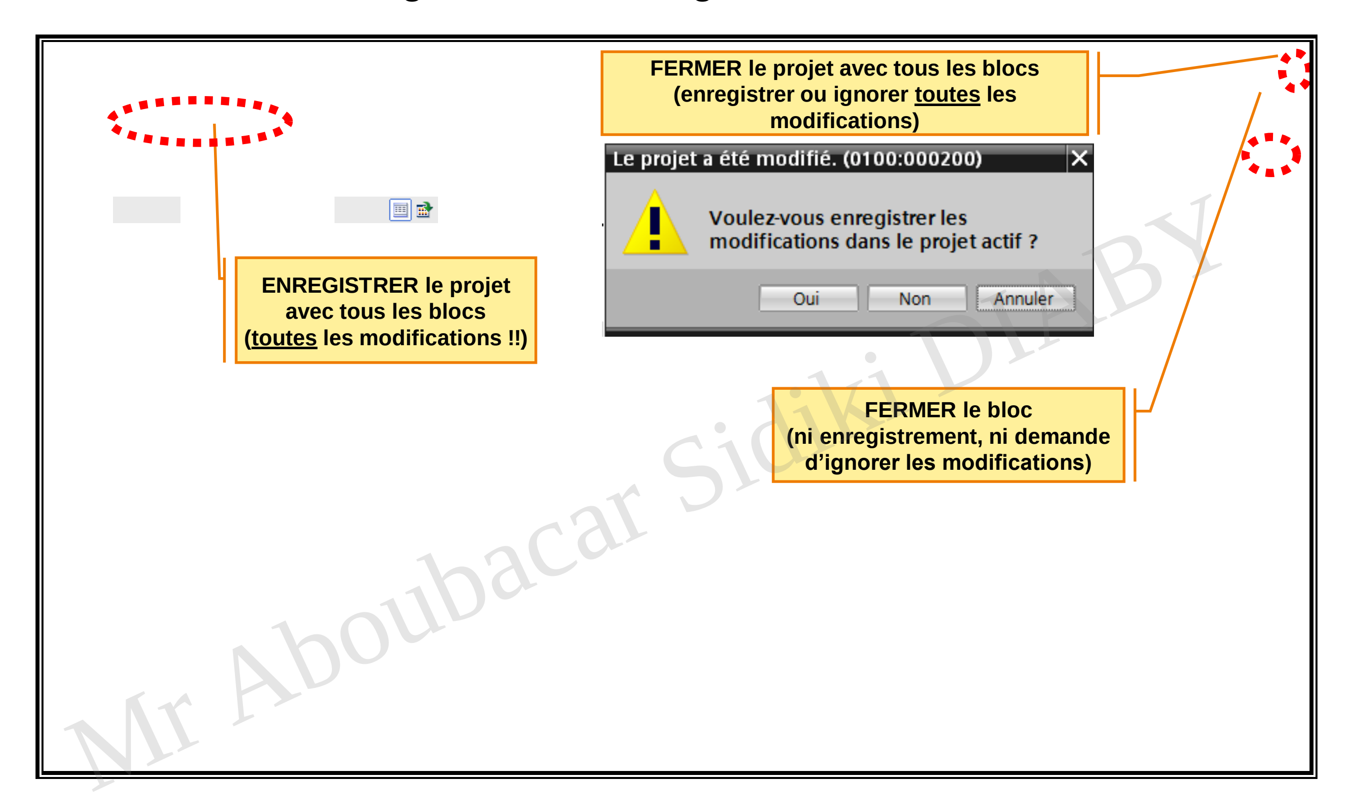
Déclarer des variables lors de la programmation

Si vous utilisez des variables inconnues lors de la programmation, vous pouvez les définir a posteriori dans le réseau.

Avantage :

La boîte de dialogue qui s’ouvre ne propose que des adresses inutilisées jusqu’ici. Cela évite par ex. des erreurs comme l’utilisation de bits appartenant à un mot déjà utilisé (accès chevauchants).

6.6. Fermer / enregistrer un bloc, ignorer les dernières modifications



Fermer un bloc

Un clic sur le bouton ☒ dans la barre de titre permet uniquement de fermer le bloc. Dans ce cas, les modifications ne sont ni ignorées ni enregistrées sur le disque dur !

Enregistrer le projet

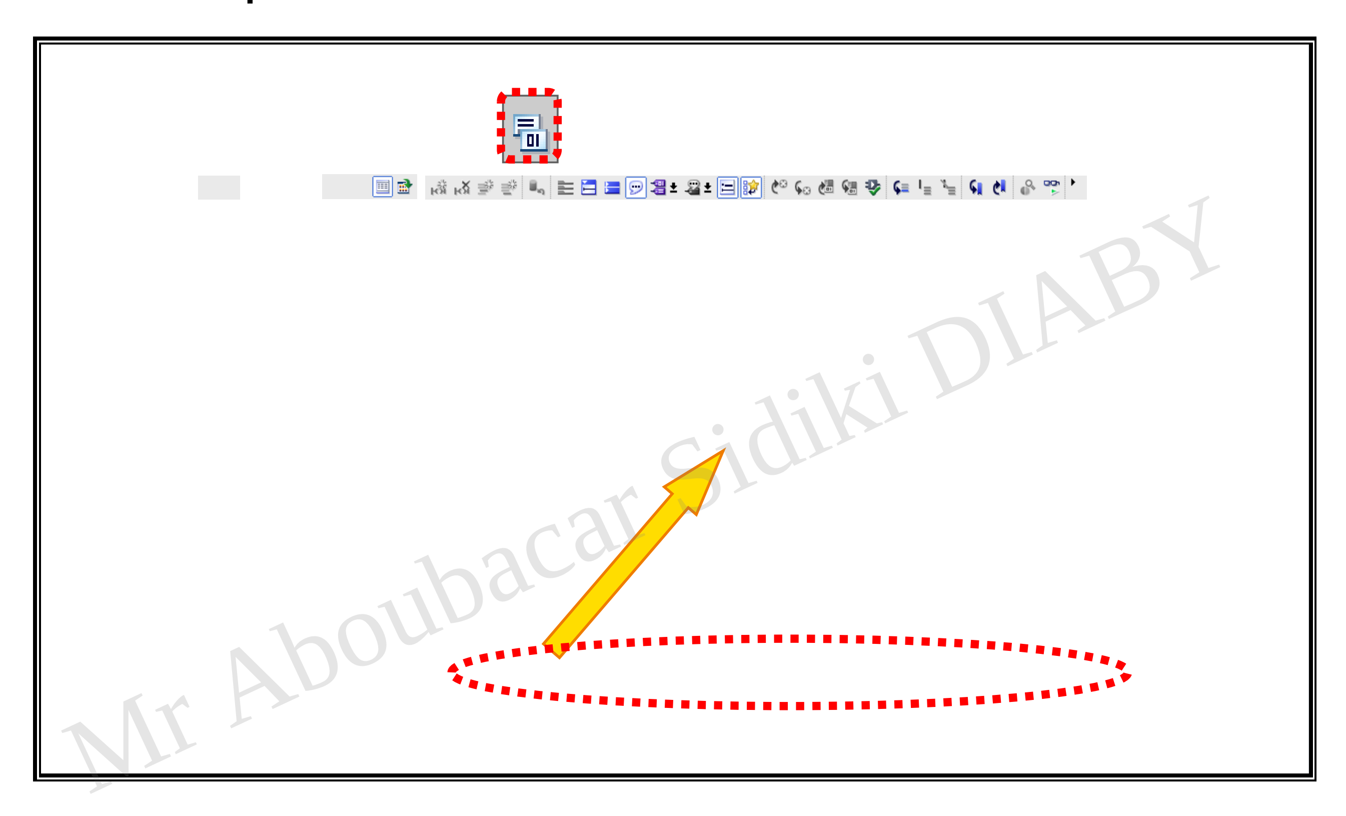


En cliquant sur « Enregistrer le projet », l'ensemble du projet (et donc également le bloc) est enregistré sur le disque dur. Toutes les modifications effectuées dans le projet sont enregistrées.

Ignorer les modifications

Pour ignorer les modifications effectuées dans le bloc, il faut fermer l'ensemble du projet sans l'enregistrer. Toutes les modifications effectuées dans le projet sont alors ignorées.

6.7. Compiler un bloc



Compilation

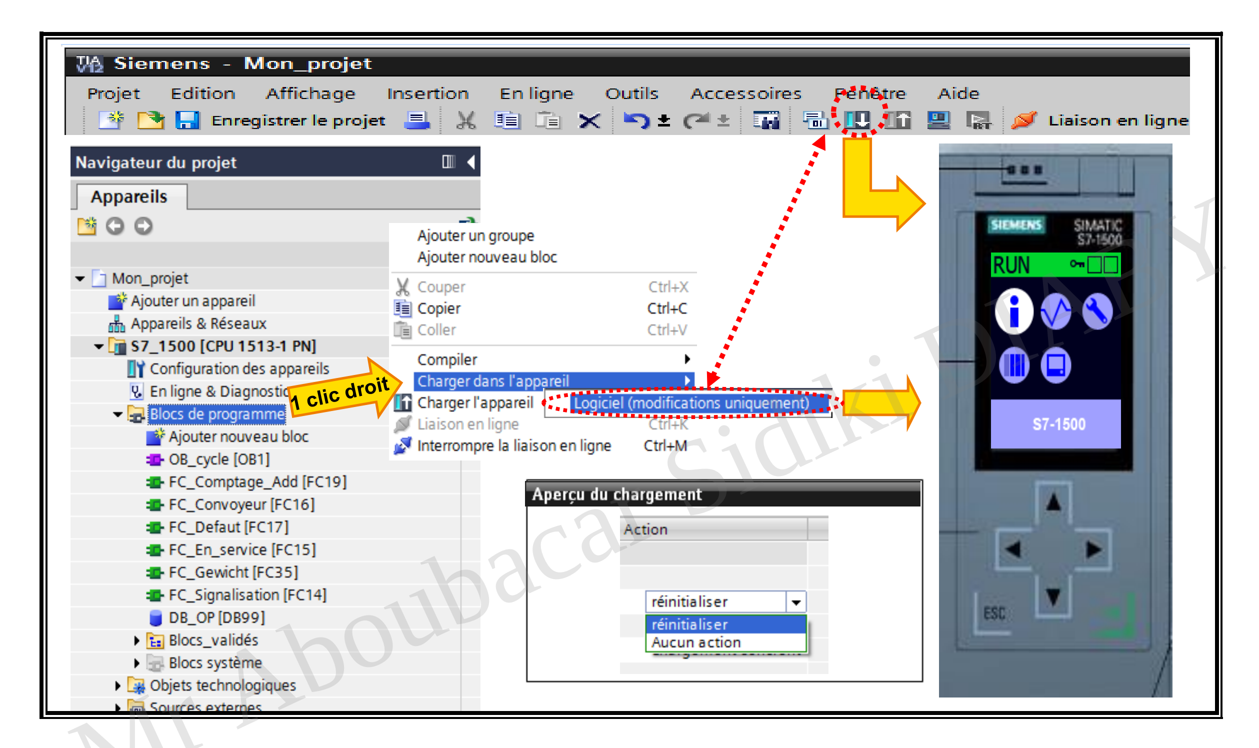
Le système compile ce qui est sélectionné dans le navigateur du projet (seul le « FC_Convoyeur » (FC16) sera compilé dans l'exemple ci-dessus). Il est possible de compiler des blocs individuels, le programme complet (« Blocs de programme » sélectionné) ou la station complète avec ses composantes logicielles et matérielles (« Station » sélectionnée).

Les menus offrent les options suivantes :

- Tout : toutes les modifications matérielles et logicielles
- Matériel : toutes les modifications matérielles
- Logiciel : toutes les modifications logicielles
- Logiciel (compilation complète des blocs) : tous les blocs

Vous pouvez contrôler l'état de la compilation dans la fenêtre d'inspection sous « Info -> Compiler ». Si des erreurs sont survenues lors de la compilation, vous pouvez directement sauter à l'emplacement de l'erreur en double-cliquant sur l'erreur indiquée.

6.7.1. Charger des blocs dans la CPU



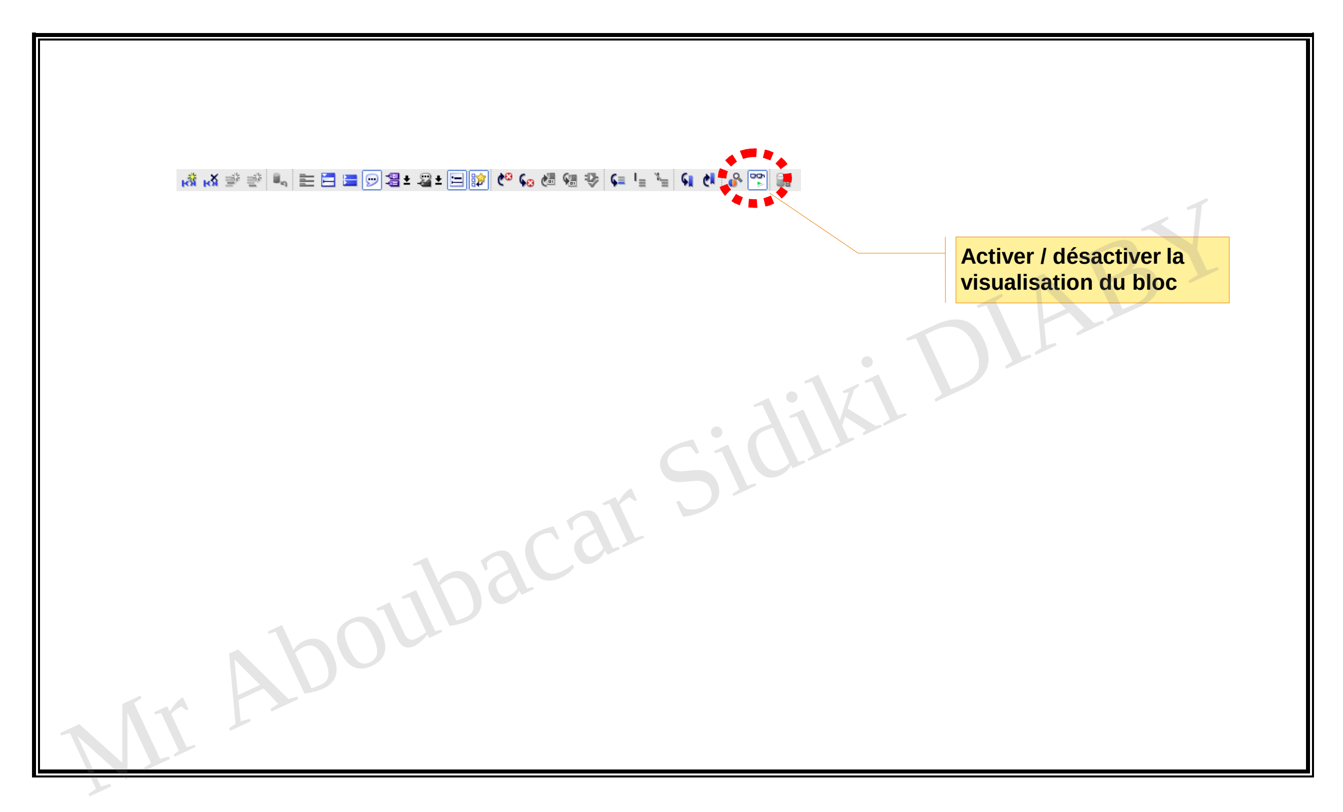
Données de projet

Les données de projet matérielles sont générées lors de la configuration du matériel, des réseaux et des liaisons. Lors du premier chargement, elles sont entièrement chargées. Lors des chargements suivants, seules les modifications sont chargées

Remarque:

Dans le menu contextuel de l'appareil (CPU) il existe également une option « Logiciel (chargement complet) » avec laquelle tous les blocs sont chargés dans la CPU même s'ils s'y trouvent déjà. Il faut tenir compte du fait que la CPU sera arrêté dans ce cas.

6.7.2. Visualiser un bloc



Visualiser un bloc

La fonction de test « Visualiser un bloc » permet de suivre l'exécution du programme au sein d'un bloc. L'écran affiche l'état ou le contenu des opérandes utilisés dans le bloc au moment de l'exécution du programme.

Visualiser

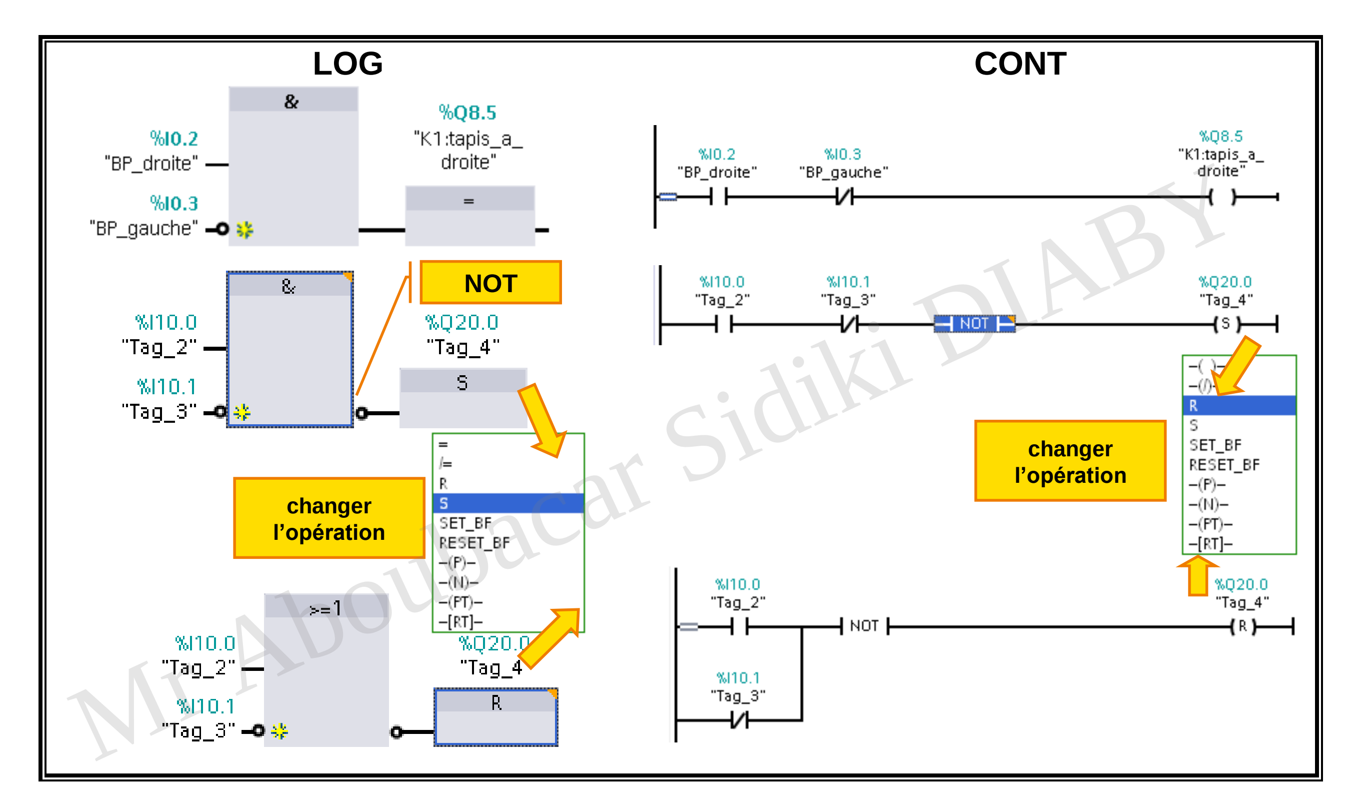
Les blocs ne peuvent être visualisés que si une liaison en ligne avec la CPU est établie. Le bloc hors ligne doit, en outre, être identique au bloc en ligne. Si le bloc ouvert hors ligne ne correspond pas au bloc mémorisé en ligne dans la CPU, vous devez, avant la visualisation, soit ouvrir le bloc mémorisé en ligne, soit charger le bloc ouvert hors ligne dans la CPU, puis le visualiser.

En mode test, les états des opérandes et les éléments CONT/LOG sont représentés avec différentes couleurs.

Exemples:

- Condition remplie → « l'élément est représenté en couleur verte »
- Condition non remplie → « l'élément est représenté en couleur bleue »

6.8. Affectation, mise à 1, mise à 0



Affectation

L'affectation transmet toujours le RLG courant à l'opérande indiqué. Le RLG affecté reste disponible après l'affectation et peut être affecté à un autre opérande ou relié à d'autres opérations.

Mise à 1

Si RLG = « 1 », l'opérande indiqué est mis à 1, si RLG = « 0 », l'état de l'opérande reste inchangé.

Mise à 0

Si RLG = « 0 », l'opérande indiqué est mis à 0, si RLG = « 0 », l'état de l'opérande reste inchangé.

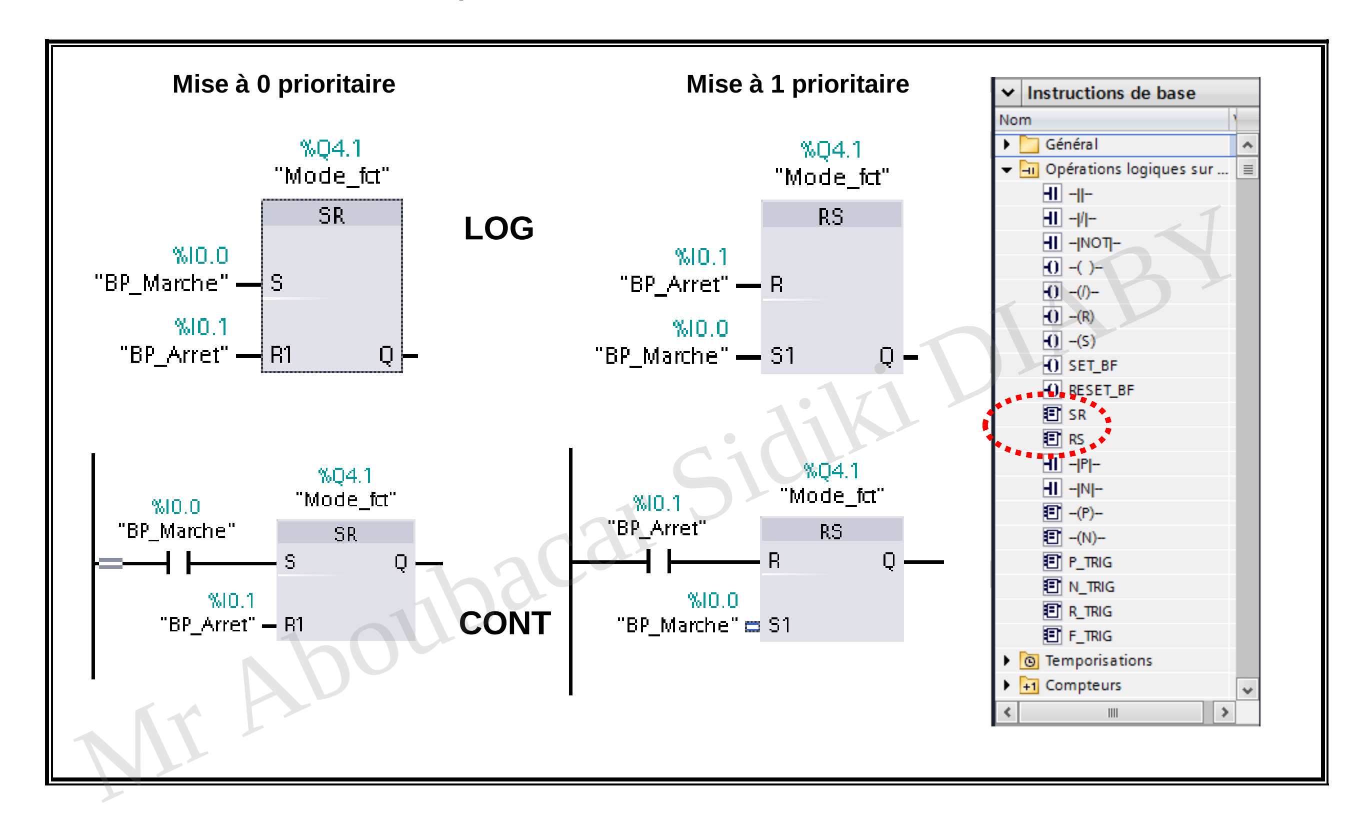
NOT

L'opération NOT inverse le RLG.

Dans l'exemple ci-dessus, si le RLG de l'opération ET est égal à « 1 », l'opération NOT l'inverse à « 0 » et l'opération de mise à 1 n'est pas exécutée (l'état de « Tag_4 » (Q20.0) reste alors inchangé).

Si le RLG de l'opération ET est égal à « 0 », l'opération NOT l'inverse à « 1 » et l'opération de mise à 1 est exécutée (« Tag_4 » (Q20.0) est mis à « 1 »).

6.8.1. Bascules mise à 1 / mise à 0



Bascule

Une bascule possède une entrée de mise à 1 et une entrée de mise à 0. Selon l'entrée à laquelle un RLG = « 1 » est présent, l'opérande est mis à 1 ou à 0.

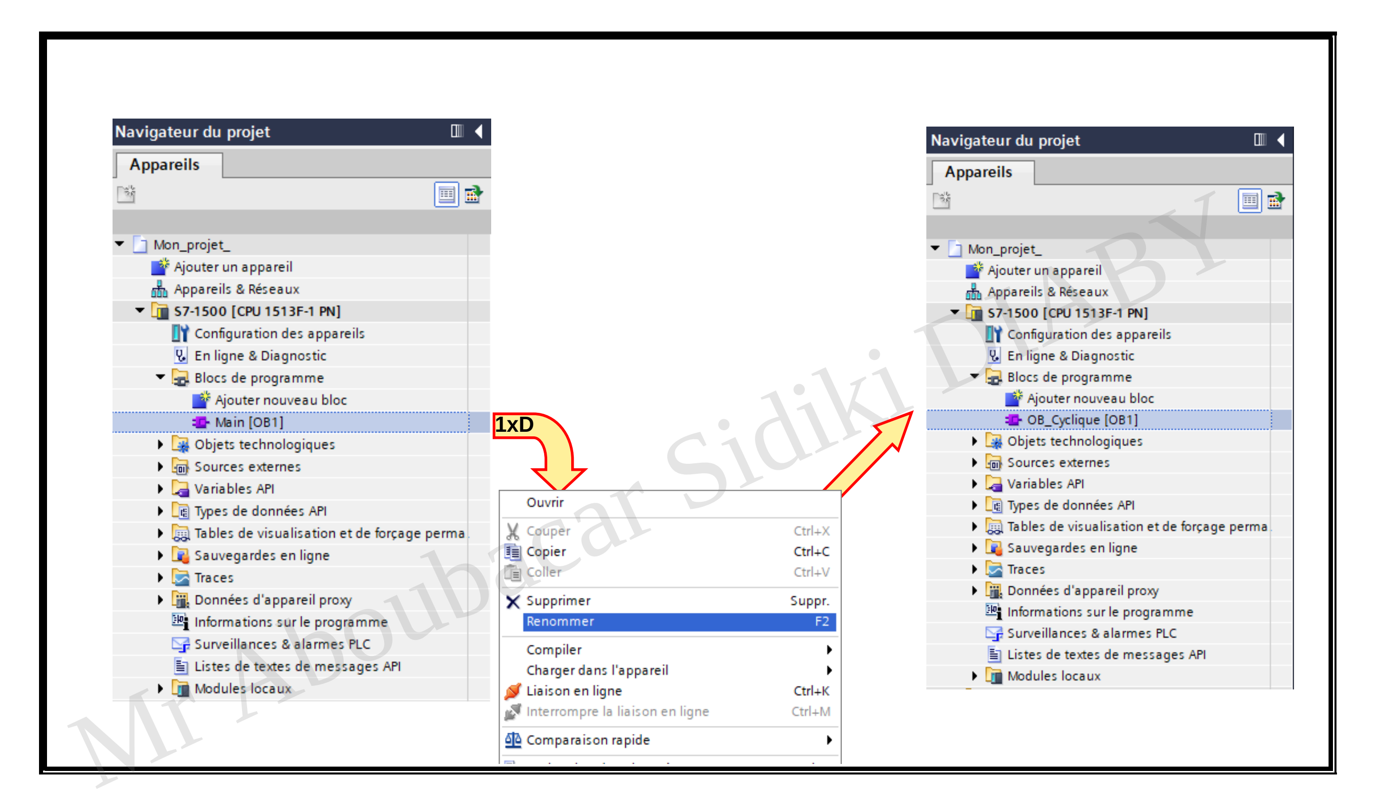
Priorité

Si les deux entrées présentent simultanément un état de signal ou un RLG = « 1 », la priorité de l'opération de mise à 1 ou de mise à 0 sera déterminante. En CONT et LOG, on dispose de ce fait de symboles différents pour la mise à 1 prioritaire et pour la mise à 0 prioritaire.

Remarque

En cas de redémarrage de la CPU, toutes les sorties sont remises à zéro ou leur état écrasé par un « 0 ».

6.9. Exercice 1: Renommer le « Main »



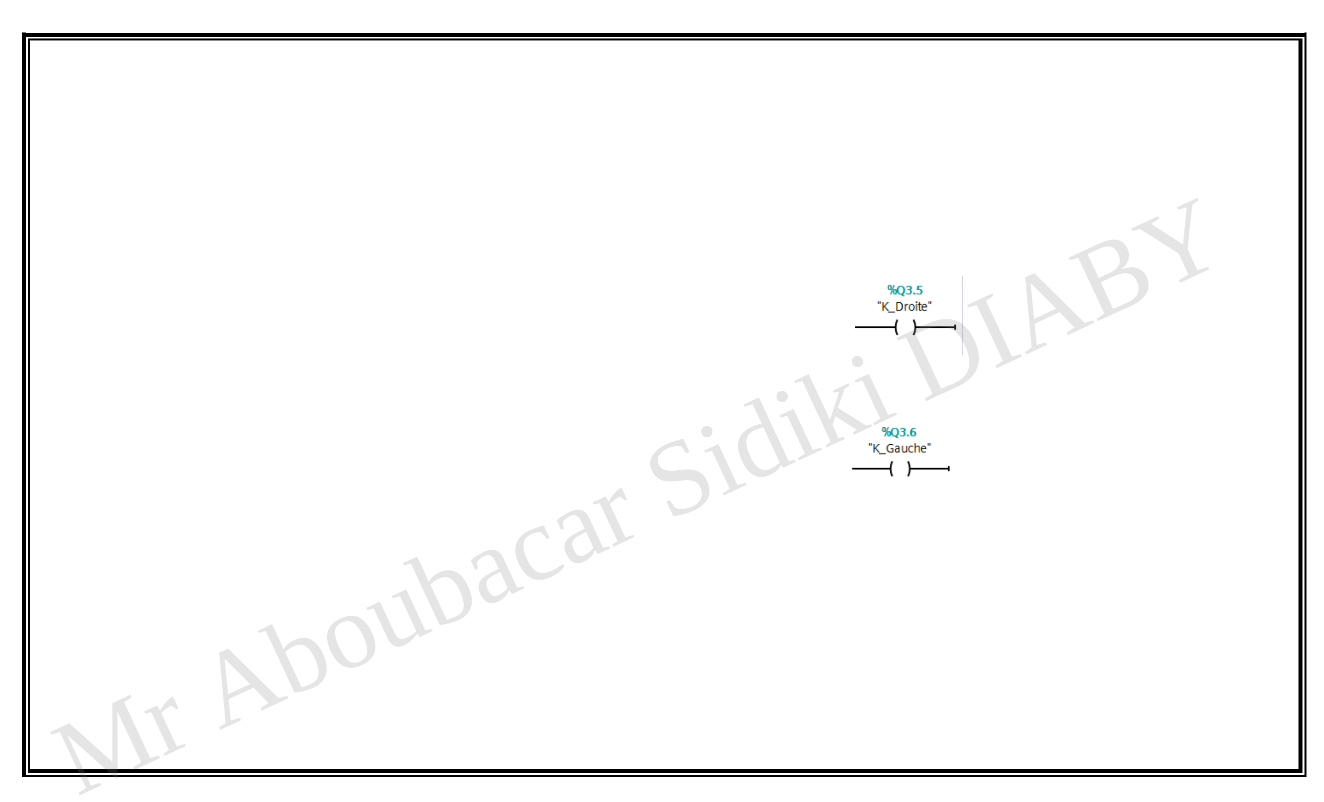
Enoncé

Vous allez renommer le bloc d'organisation.

Marche a suivre

1. Ouvrez « Blocs de programme ».
2. Ouvrez le menu contextuel du bloc « Main » par clic droit.
3. Sélectionnez « Renommer ».
4. Donnez le nom symbolique « OB Cyclique ».
5. Enregistrez votre projet.

6.9.1. Exercice 2 : Programmation du mode manuel



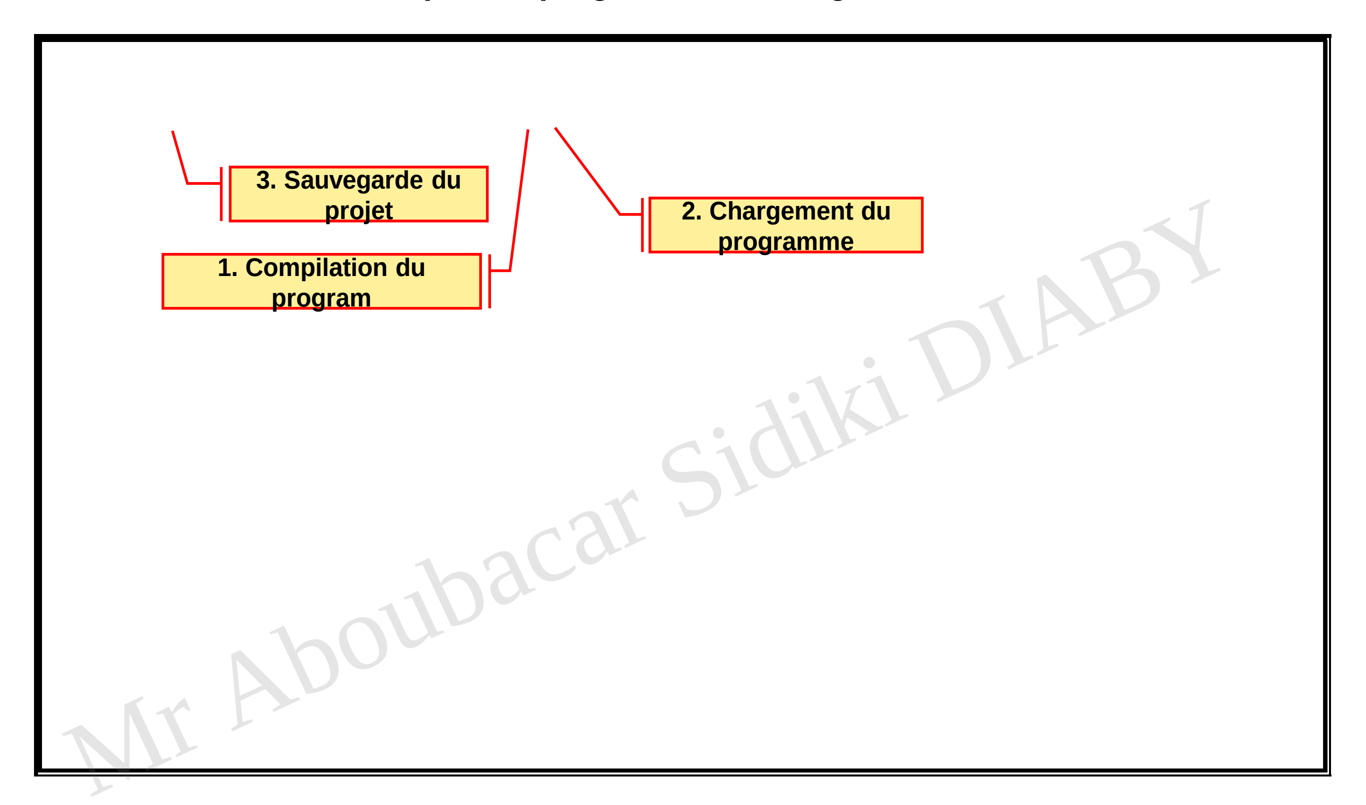
Enoncé

Programmez la marche par impulsions du convoyeur comme indiqué dans la fenêtre.

Marche à suivre

1. Programmez une opération logique ET avec un « contact NO » et un contact « NF » à partir des « Favoris » vers le réseau par glisser - déposer.
2. Programmez une assignation à partir de la barre des onglets « Instructions » vers le réseau comme indiqué ci dessus.
3. Affectez « BP_droite » au premier opérande de l'opération logique ET (I 0.2), (Vous pouvez saisir un nom symbolique ou une adresse absolue). Procédez de manière identique pour « BP_gauche » (I 0.3).
4. Dans le navigateur du projet, sélectionnez (ne pas ouvrir) la table des variables « Convoyeur » et glissez « K_Droite » (Q3.5) à partir de la « Vue détaillée » en tant qu'opérande de l'assignation.
5. Attribuez un titre au réseau.
6. Insérez un nouveau réseau et programmez l'opération logique correspondante pour piloter le convoyeur vers la gauche.
7. Fermez le bloc d'organisation.
8. Enregistrez votre projet.

6.9.2. Exercice 3: Compilez le programme et chargez le dans la CPU



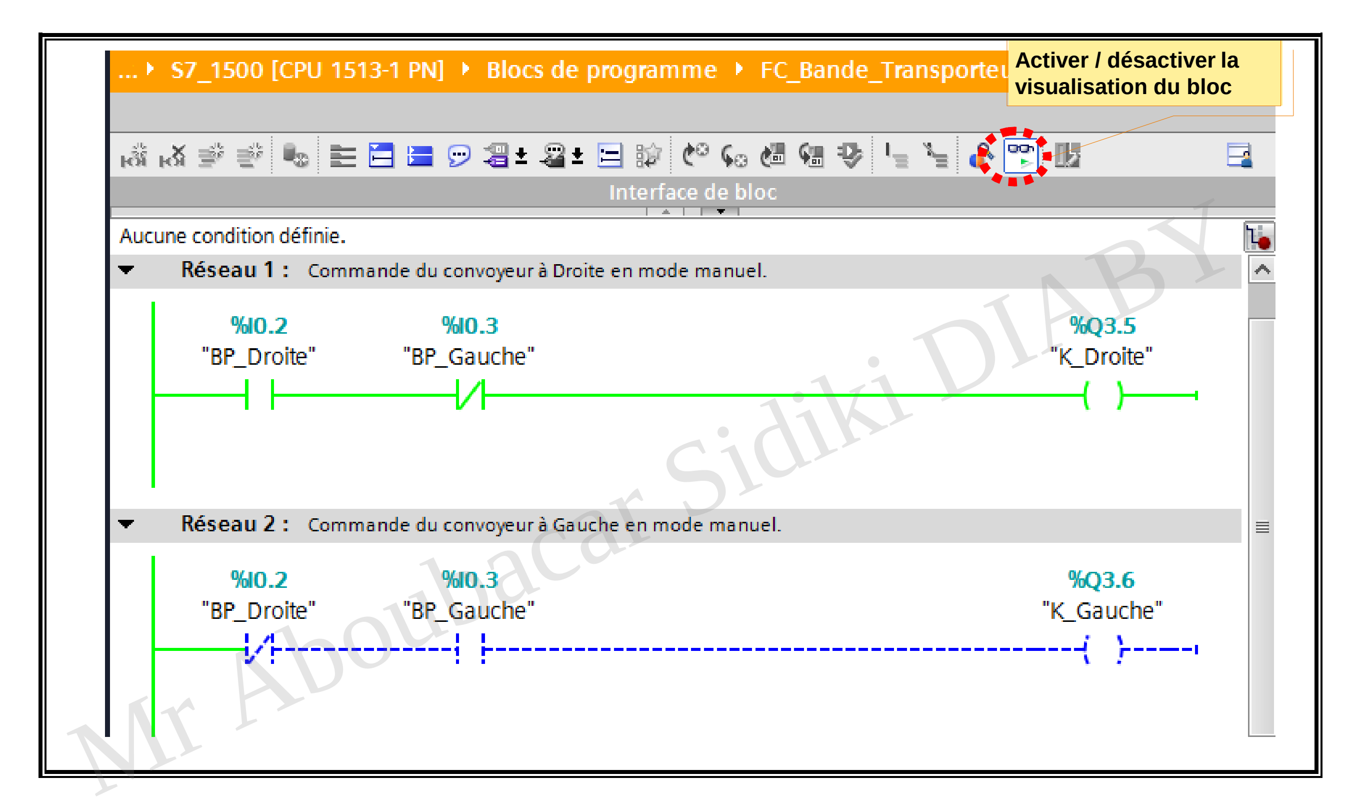
Enoncé

Vous allez compiler, charger et enregistrer le projet.

Marche à suivre

1. Sélectionnez « Blocs de programme ».
2. Compilez votre programme avec le bouton « Compiler » dans la barre des tâches.
3. Chargez votre programme avec le bouton « Charger dans l'appareil » dans la barre des tâches.
4. Enregistrez votre projet.

6.9.3. Exercice 4: Visualiser un bloc



Enoncé

Vous voulez visualiser le programme en ligne.

Marche à suivre

1. Ouvrez « OB_Cyclique ».
2. Visualisez le bloc (le programme) avec le bouton « Activer/désactiver visualisation du programme ».
3. Actionnez les boutons « BP_Droite » et « BP_Gauche » du simulateur.

Sommaire 7

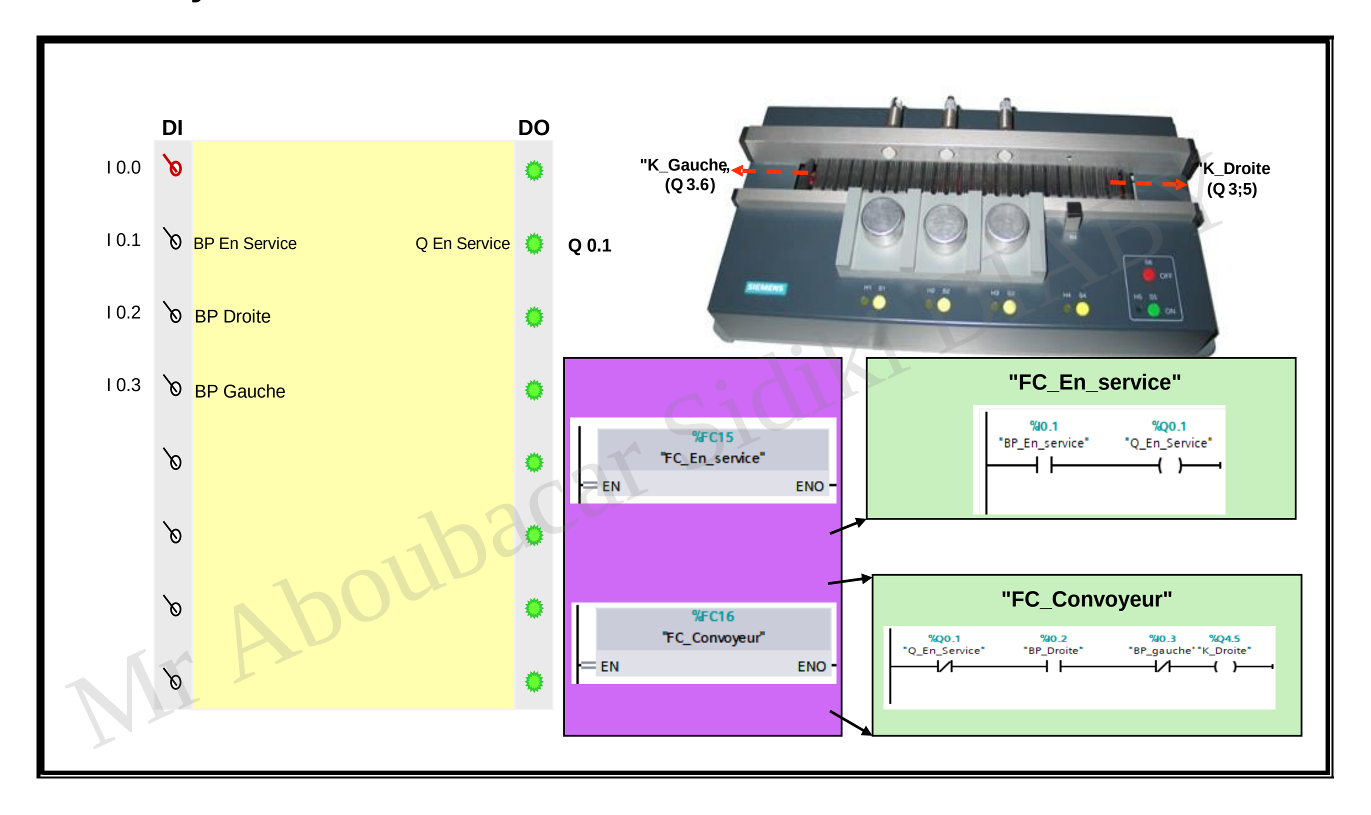
7. Blocs de programme et éditeur de blocs

A l'issue du chapitre, vous allez ...

- ... connaître les différents types de blocs S7
- ... connaître le principe de la programmation structurée
- … Ajouter/créer un nouveau bloc (Fonction)
- ... Modifier les propriétés d'un bloc
- … programmer l'appel d'un bloc
- … connaitre la procédure de re-chargement d'un bloc
- … être familiariser avec le concept mémoire du S7-1500



7.1. Description de l'installation: Le convoyeur à bande est un système de distribution avec mode de sélection



.

Modes de fonctionnement

Le bouton « BP En Service », permet d'activer le mode « Q En Service ».

Hors Service (« Q En Service »)

Lorsque l'installation est « Hors Service », la bande peut être commandée manuellement vers la droite via le bouton « BP Droite », et vers la gauche via le bouton « BP Gauche ». Si les 2 boutons sont activé , appuyés simultanément le convoyeur ne doit pas bouger.

7.2. Aperçu des blocs dans STEP 7

Bloc / instructions		Propriétés
Bloc d'organisation (OB)		- Interface utilisateur - Priorités échelonnées (1..27) - Informations spécifiques de traitement dans la pile de données locale
Fonction (FC)		- paramétrable (les paramètres doivent être affectés lors de l'appel) - Sans mémoire (uniquement variables temporaires)
Bloc fonctionnel (FB)		- paramétrable (les paramètres peuvent être affectés lors de l'appel) - Avec mémoire (variables statiques)
Blocs de données (DB)		- Stockage structuré des données locales (DB d'instance) - Stockage structuré des données globales (dans tout le programme)
Fonction système(SFC)	Instruction	- Bloc de fonction stocké dans le système d'exploitation de la CPU pouvant être appelé par l'utilisateur (sans mémoire)
Bloc fonctionnel Système (SFB)	Instruction avec Instance	- Bloc fonctionnel stocké dans le système d'exploitation de la CPU pouvant être appelé par l'utilisateur (avec mémoire)
System data block (SDB)		- Blocs de données pour les données de configuration et paramètres

Blocs

Le système d'automatisation offre différents types de bloc dans lesquels peuvent être mémorisés le programme utilisateur et les données correspondantes. Le programme peut être structuré en différents blocs selon les exigences du processus à automatiser. L'ensemble des opérations disponibles est utilisable par tous les blocs (FB, FC et OB).

Blocs d'organisation (OB)

Les blocs d'organisation (OB) constituent l'interface entre le système d'exploitation et le programme utilisateur.

Fonctions (FC)

Les fonctions (FC) contiennent une fonctionnalité partielle du programme. Elles peuvent être programmées de manière à être paramétrables. Elles reçoivent alors les paramètres nécessaires lors de leur appel. Elles conviennent de ce fait à la programmation de fonctionnalités partielles complexes et récurrentes, comme les opérations de calcul.

Blocs fonctionnels (FB)

Les blocs fonctionnels offrent les mêmes possibilités que les fonctions (FC), mais ils disposent en plus de zones de mémoire propres (blocs de données d'instance). Ils conviennent de ce fait à la programmation de fonctionnalités plus complexes et récurrentes, comme les tâches de régulation.

Mr Aboubacar Sidiki DIABY

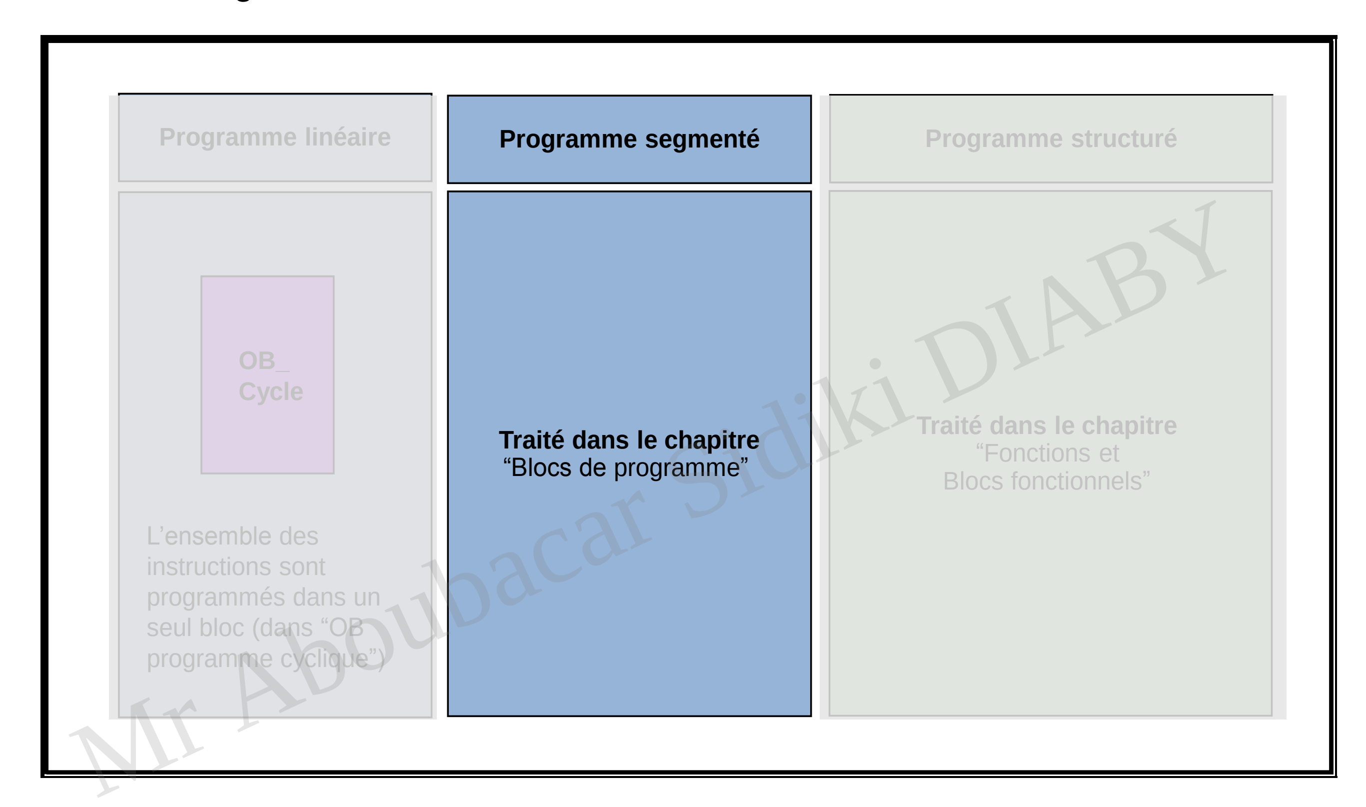
Blocs de données:

Blocs de données are used to store data. There are global Blocs de données and instance Blocs de données which are used for the data backup of an FB call.

System Instructions:

System instructions are used to solve frequently required standard tasks. They are integrated in the CPU's operating system and therefore do not require any additional memory.

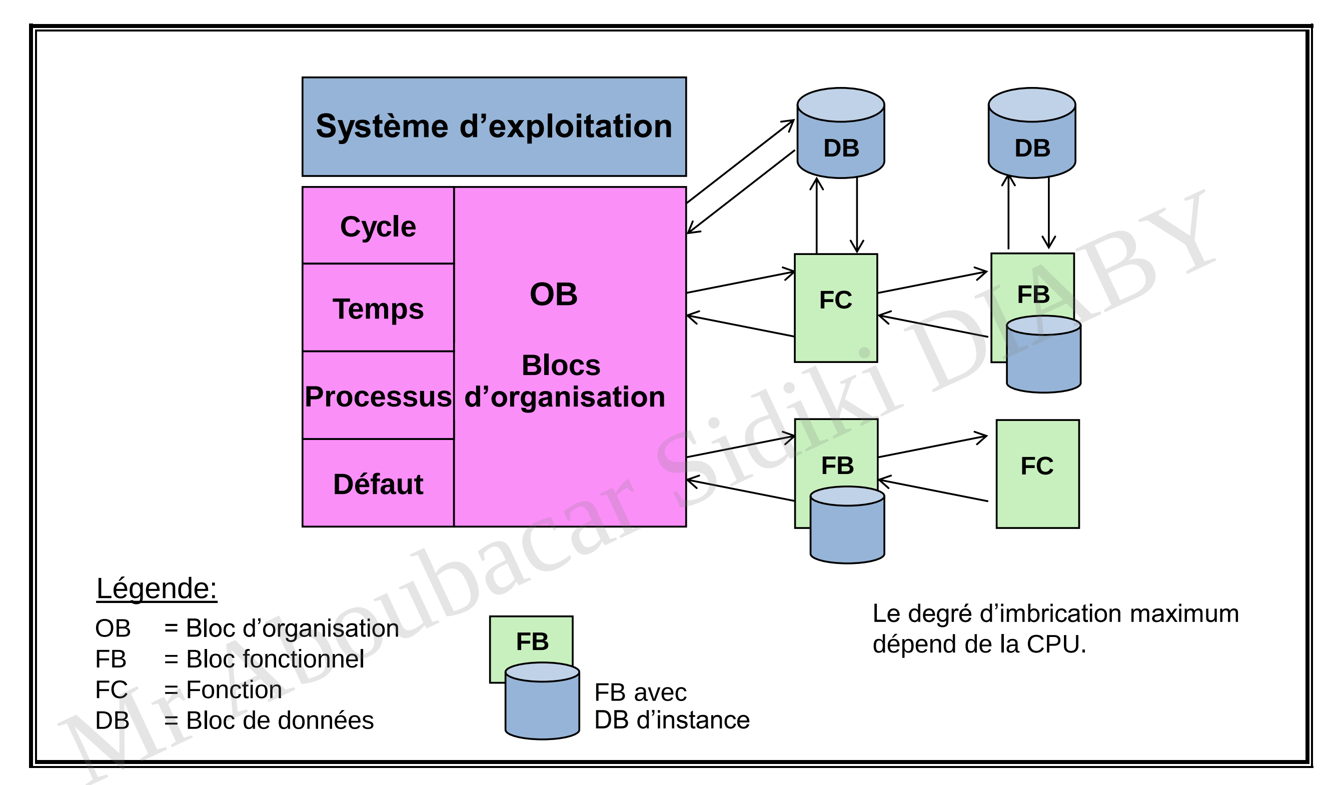
7.2.1. Programmation structurée



Programme segmenté

Le programme est segmenté en plusieurs blocs, chacun contenant uniquement le programme nécessaire pour réaliser la tâche qui lui est assignée. Un bloc peut être subdivisé en plusieurs segments, appelés réseaux. Si le programme fait appel à des réseaux similaires, il est possible de créer des modèles. En général, le bloc d'organisation OB 1 contient des instructions qui appellent les autres blocs du programme dans un ordre défini.

7.2.2. Types de blocs de programme



Le système d'automatisation offre différents types de bloc dans lesquels peuvent être mémorisés le programme utilisateur et les données correspondantes. Le programme peut être structuré en différents blocs selon les exigences du processus à automatiser. L'ensemble des opérations disponibles est utilisable par tous les blocs (FB, FC et OB).

Blocs d'organisation (OB)

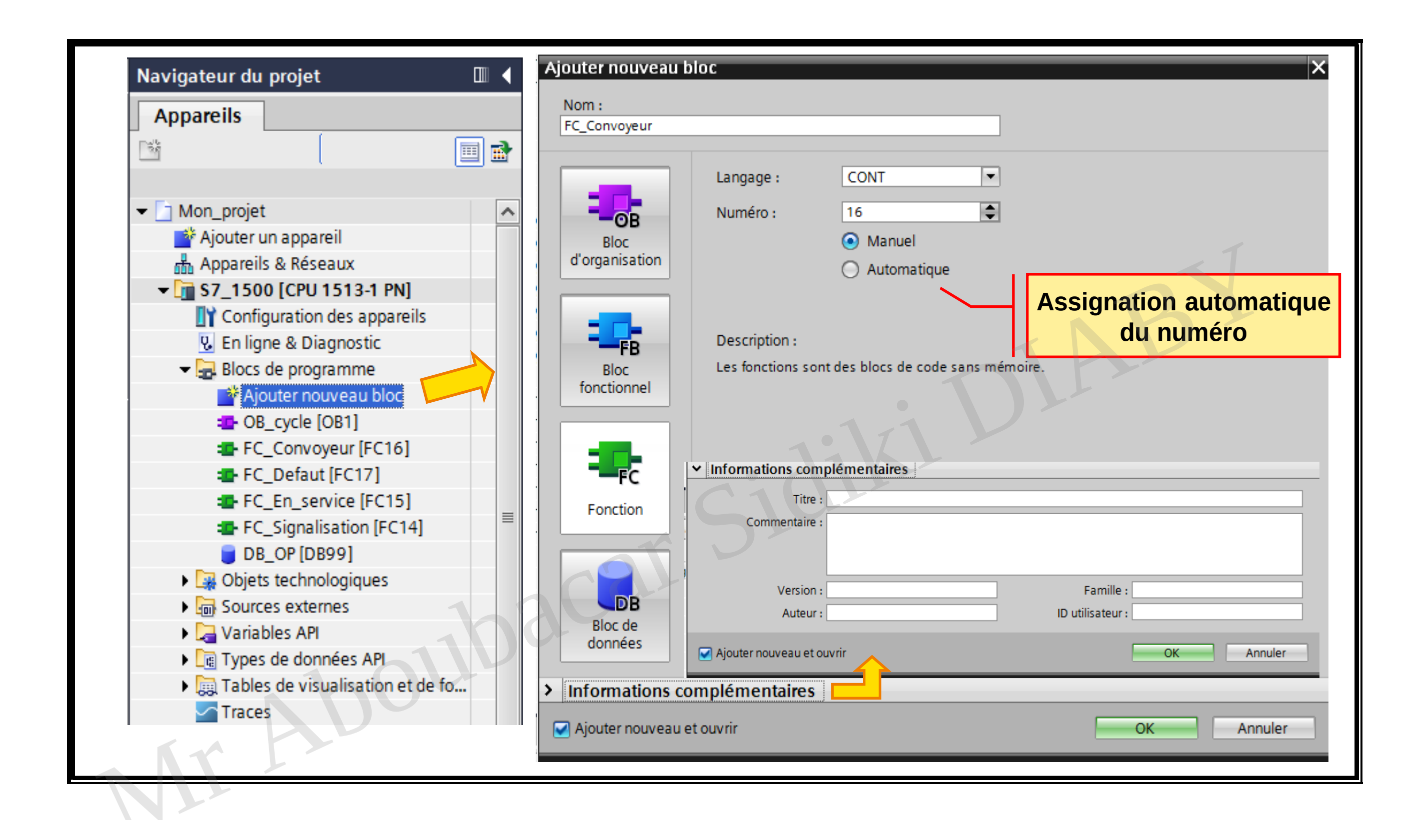
Les blocs d'organisation (OB) constituent l'interface entre le système d'exploitation et le programme utilisateur. L'ensemble du programme peut être stocké dans l'OB1 appelé de manière cyclique par le système d'exploitation (programme linéaire) ou encore réparti sur plusieurs blocs (programme structuré).

Fonctions (FC)

Les fonctions (FC) contiennent une fonctionnalité partielle du programme. Elles peuvent être programmées de manière à être paramétrables. Elles reçoivent alors les paramètres nécessaires lors de leur appel. Elles conviennent de ce fait à la programmation de fonctionnalités partielles complexes et récurrentes, comme les opérations de calcul.

Blocs fonctionnels (FB)

Les blocs fonctionnels offrent les mêmes possibilités que les fonctions (FC), mais ils disposent en plus de zones de mémoire propres (blocs de données d'instance). Ils conviennent de ce fait à la programmation de fonctionnalités plus complexes et récurrentes, comme les tâches de régulation.

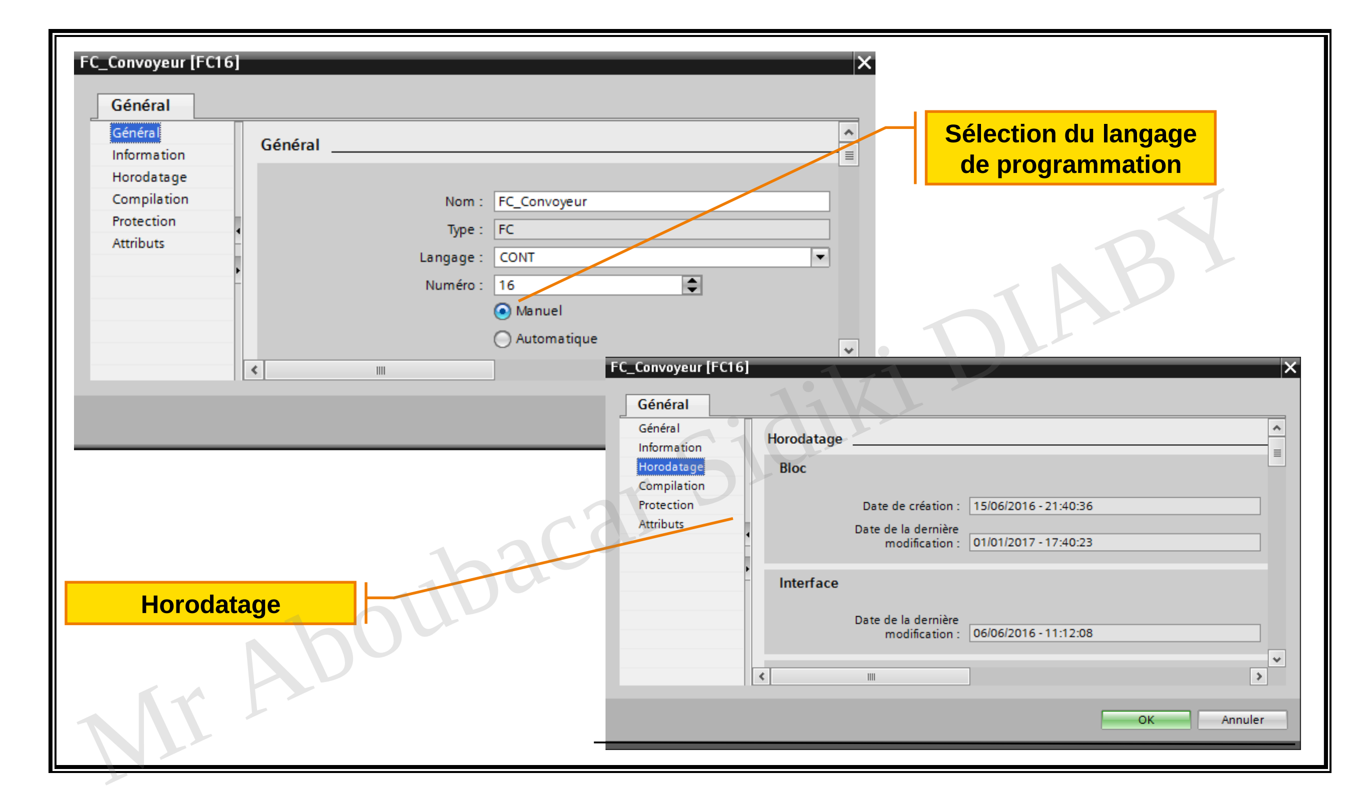


Ajouter un nouveau bloc

Pour ajouter un nouveau bloc, procédez comme indiqué sur la figure ci-dessus. Lors de la création du bloc, vous devez notamment définir le type de bloc (OB, FB, FC ou DB), le langage de programmation, le nom symbolique (mnémonique) et le numéro du bloc. Les numéros de bloc peuvent être attribués automatiquement ou manuellement.

« Informations complémentaires » vous permet de documenter plus en détail le bloc, notamment en indiquant un numéro de version et un auteur.

7.3. Propriétés du bloc : langage de programmation, horodatage

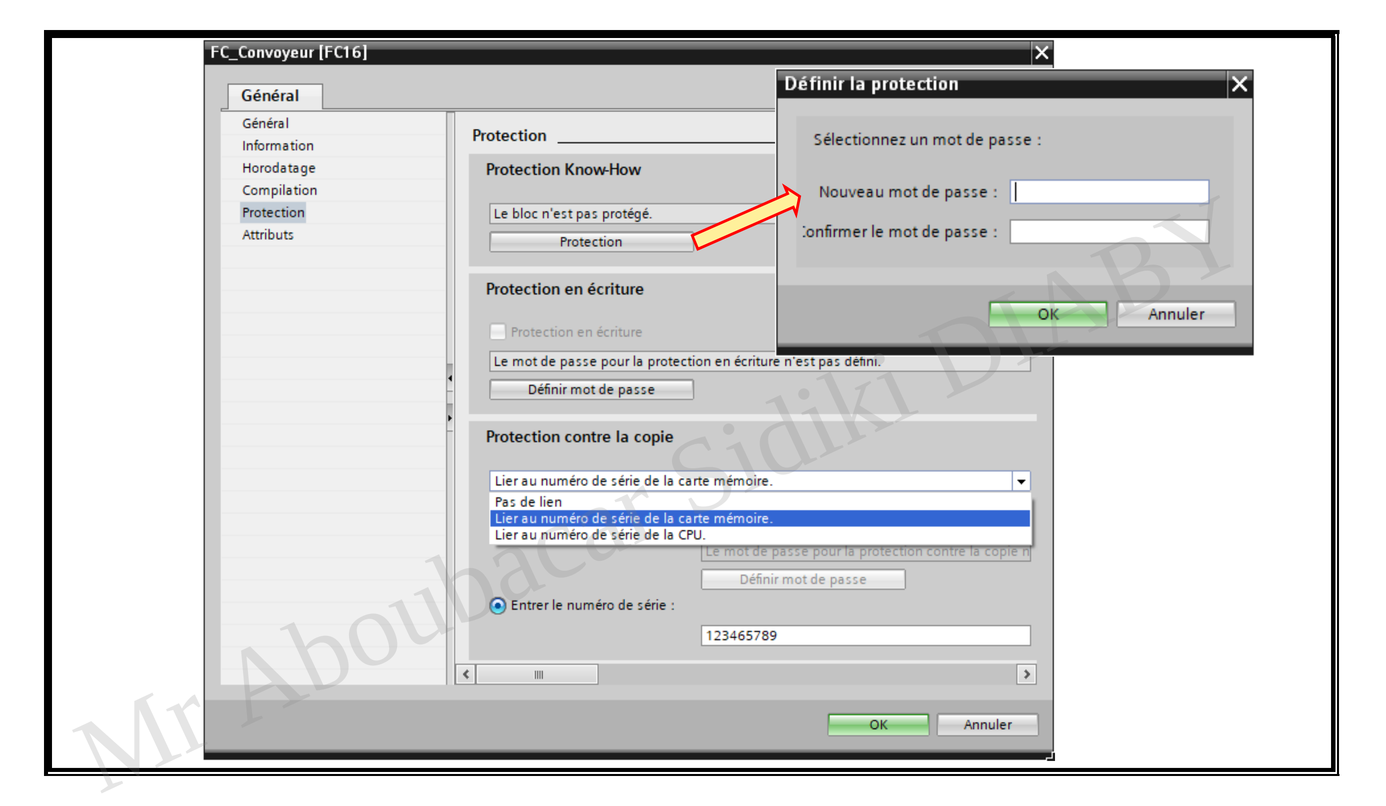


Propriétés

Chaque bloc possède des propriétés spécifiques qui peuvent être affichées et éditées. Ces propriétés assurent les tâches suivantes :

- Identification du bloc
- Affichage de l'espace mémoire requis et de l'état de la compilation du bloc
- Affichage de l'horodatage
- Affichage des Informations de référence
- Définition de la protection contre les accès non autorisés
- Affichage et changement du langage de programmation

7.3.1. Propriétés du bloc : protection du savoir-faire (know-how)



Protection du savoir-faire (« Protection Know-How »)

Les blocs peuvent être protégés par un mot de passe contre les accès non autorisés. Sur les blocs avec protection du savoir-faire, seules les données suivantes sont accessibles en lecture :

- Paramètres de transfert Input, Output, InOut, Return
- Titre du bloc
- Commentaire du bloc
- Propriétés du bloc
- Références croisées avec indication des variables globales utilisées

Comme les blocs non protégés, les blocs avec protection du savoir-faire peuvent être copiés, appelés, chargés dans la CPU et effacés. Le code du bloc est en revanche protégé contre l'écriture et les modifications non autorisées.

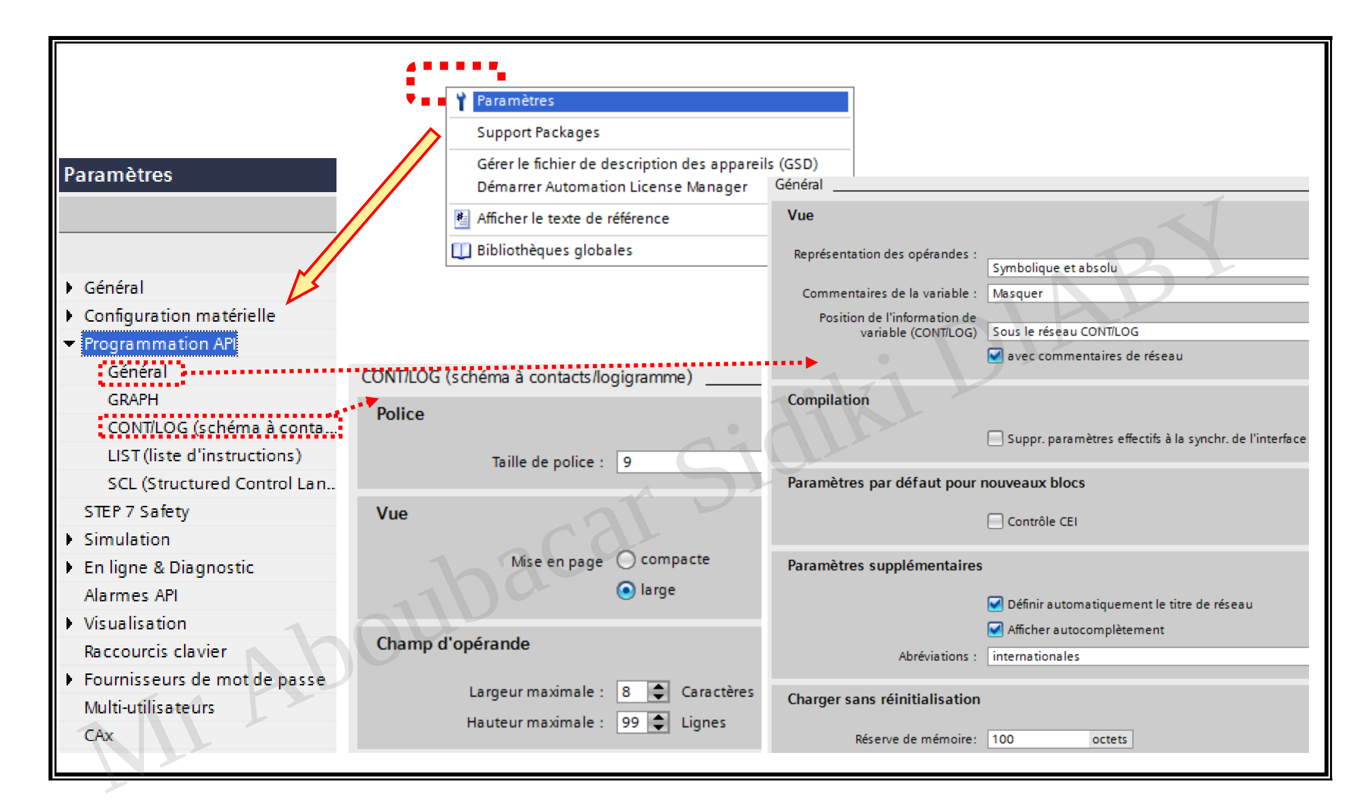
Protection contre la copie

Sur le S7-1200/1500, il est possible d'activer une protection contre la copie en plus de la protection du savoir-faire, Elle permet de définir la carte mémoire ou la CPU (identifiées par leur numéro de série) sur laquelle le bloc peut être copié.

Attention !

En cas d'oubli du mot de passe, l'accès au bloc devient impossible.

7.4. Paramètres de l'éditeur de blocs



Paramètres de l'éditeur de blocs

Les paramètres permettent seulement de définir la manière dont les blocs seront représentés à l'ouverture. A l'intérieur de l'éditeur, il est possible de modifier à tout moment la représentation des blocs, par ex. l'affichage ou le masquage des commentaires.

- Compilation
  Si l'option « Suppression des paramètres effectifs à la synchronisation de l'interface » est cochée, les appels de blocs paramétrés sont automatiquement adaptés en cas de suppression ultérieure de paramètres au sein du bloc.

- Contrôle CEI
  Seules les variables de type de données absolument correct peuvent être utilisées. Si une opération requiert par ex. une variable du type de données INT, il n'est pas possible d'utiliser une variable de type de données WORD, même si la taille (16 bits) est la même.

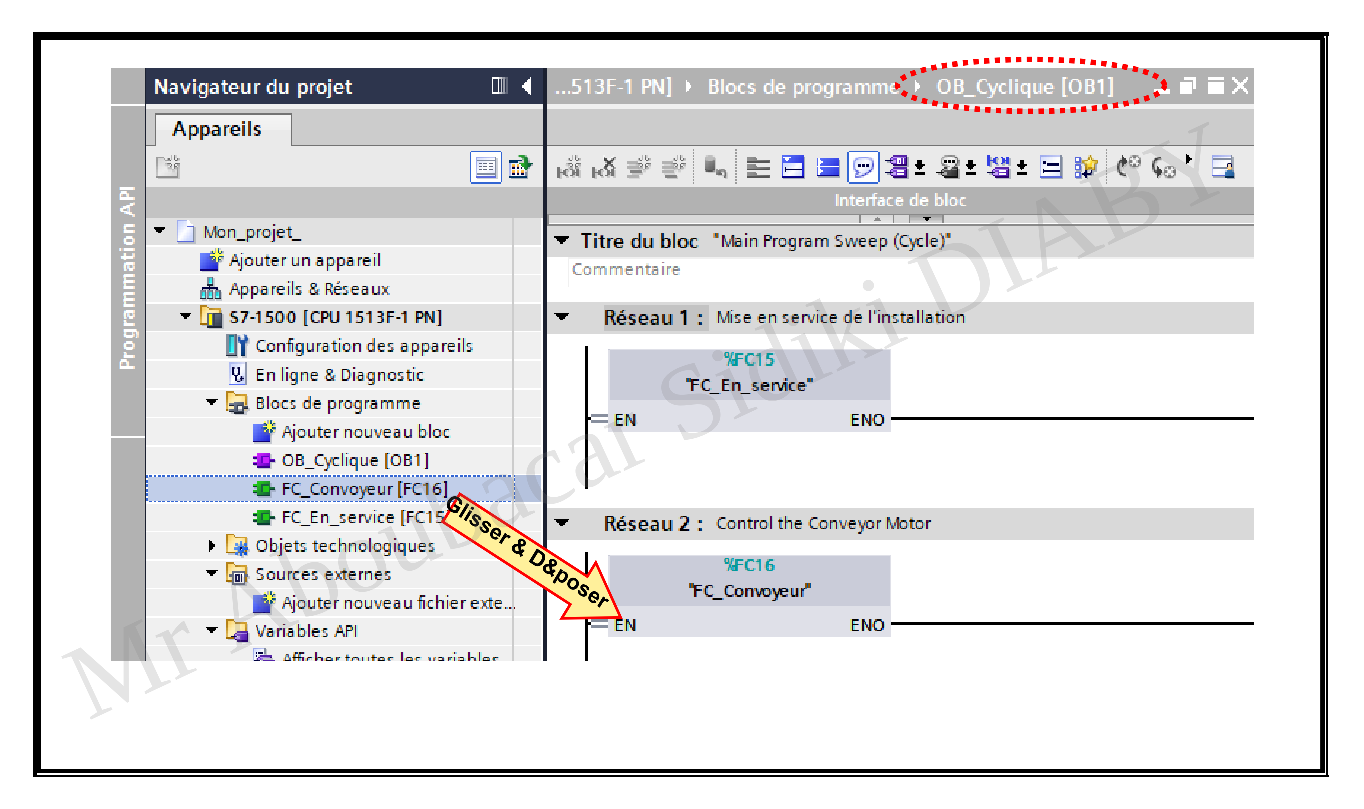
- Accès au bloc optimisé
  Les variables des blocs de données et les variables locales contenues dans les blocs ne peuvent être adressées que de manière symbolique et non absolue. Avantage : optimisation de l'occupation mémoire (sans octets de remplissage) et temps d'accès plus courts

- Abréviations
  Choix de la syntaxe du langage de programmation :
  – allemand (par ex. E pour « Eingang »)
  – internationales (par ex. I pour « Input »)

- Mise en page
  Si l'option « avec adresse absolue » est cochée, les opérandes globaux s'affichent avec leurs adresses absolues.

- Champ d'opérande
  Paramétrage de la largeur et de la hauteur maximale des symboles dans les logigrammes et les schémas à contacts

7.5. Programmation des blocs : appel d'un bloc

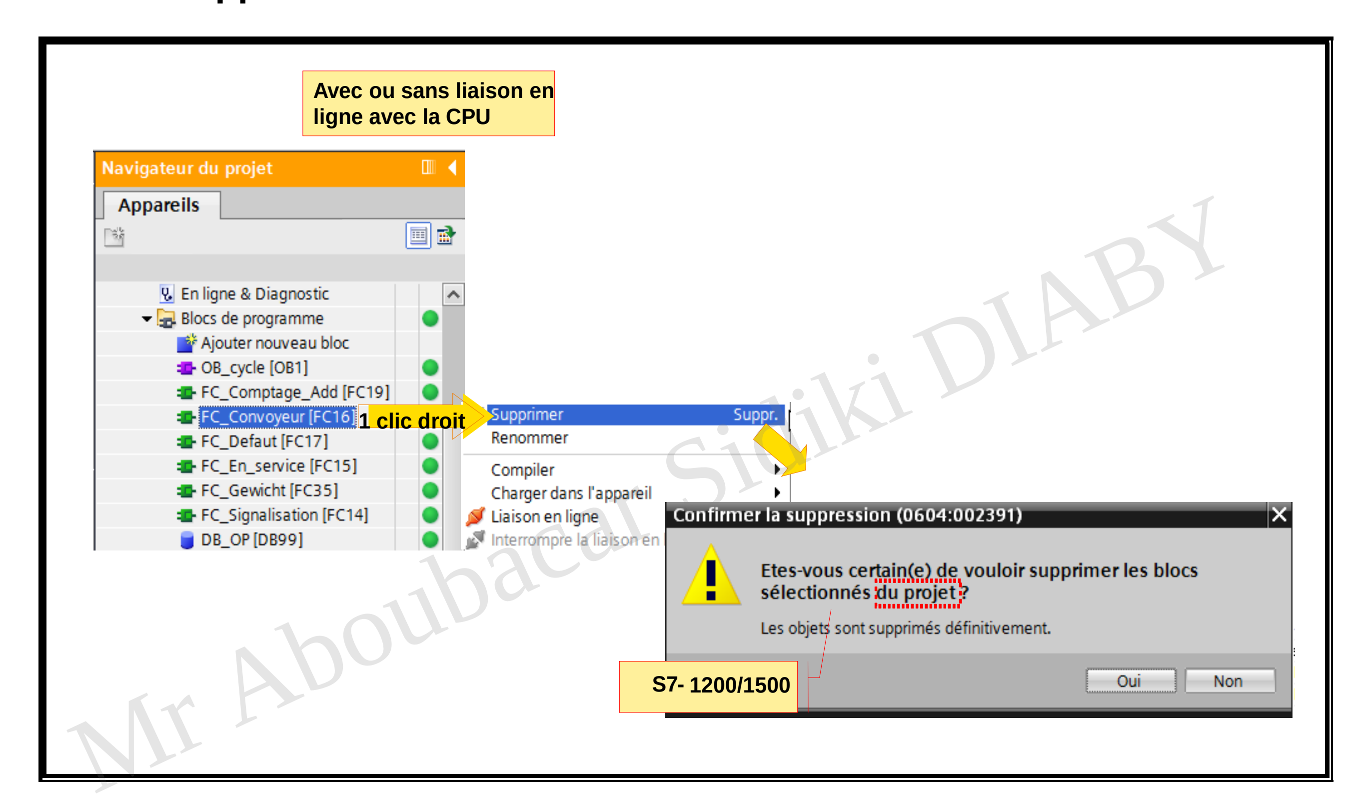


Appel d'un bloc

Lorsqu'un bloc appelle un autre bloc, les instructions du bloc appelé sont exécutées. Le traitement du bloc appelant n'est repris que lorsque le traitement du bloc appelé est terminé. Il se poursuit à l'opération qui suit l'appel du bloc.

L'appel d'un bloc peut être programmé par glisser-déposer ou par copier-coller.

7.6. Supprimer des blocs

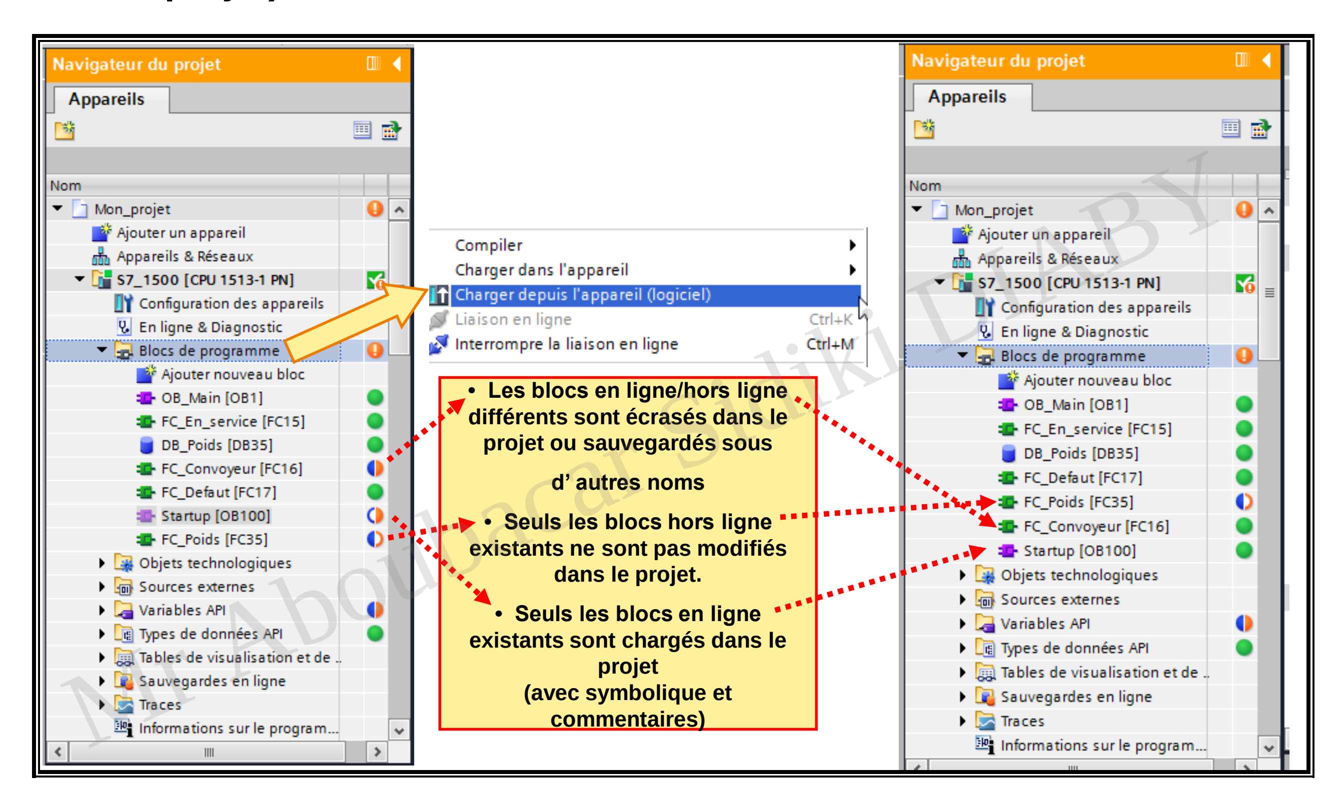


Supprimer des blocs

En ligne, les blocs <u>ne peuvent pas être effacés</u> directement dans la CPU. Si un bloc est sélectionné (même avec une connexion Enligne établie) et supprimé, il apparait la boite de dialogue pour confirmer si le bloc sélectionné peut être effacé *Horsligne*.

Par conséquent, avec un chargement complet du programme, les blocs qui n'exitent que dans la CPU sont éffacés En Ligne.

7.7. S7-1200/1500 : « Charger de l'appareil » (chargement dans le projet)



Charger des blocs dans le projet:

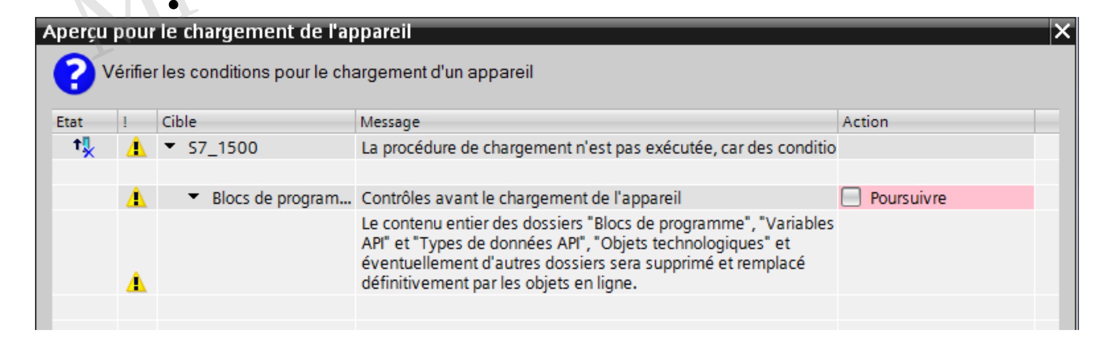
« Charger depuis l'appareil » permet de transférer depuis la CPU dans le projet des blocs individuels ou un programme complet avec les objets technologique, les tables de variables API et les types de données API.

Si le dossier des blocs ou des blocs seuls sont sélectionnés, alors...

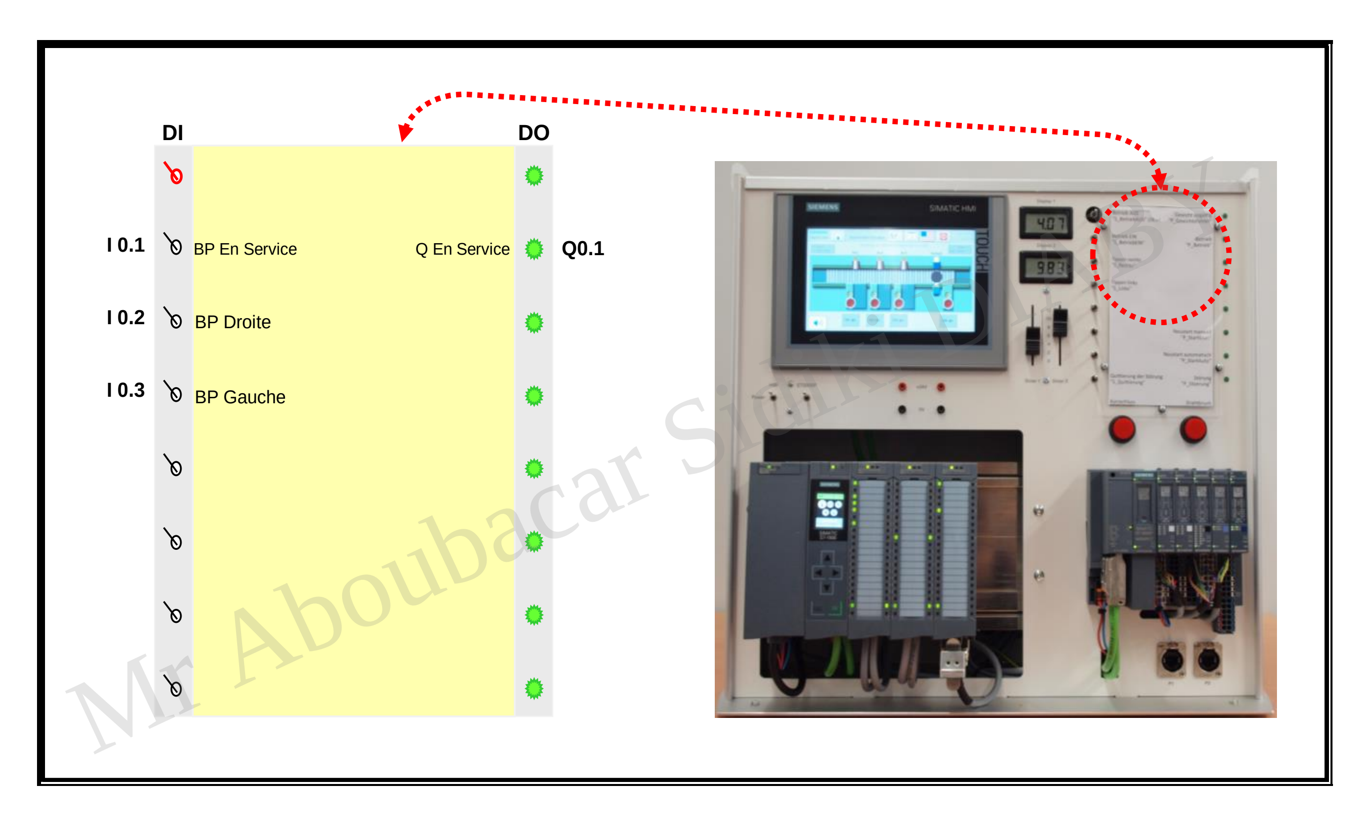
- ...les blocs qui n'existent qu'en mode Hors Ligne dans la CPU sont chargés dans le projet Hors Ligne.
- ...les blocs qui diffèrent en mode En Ligne/ Hors Ligne sont, au niveau du projet Hors Ligne, soit écrasés par les blocs transférés soit récupérés et sauvegardés sous un autre nom (mais avec le même numéro!). Ce choix est accessible au niveau du dialogue de chargement.
- ...les blocs qui n'existaient qu'en mode Hors ligne ne sont pas touchés.

Si la station est sélectionnée, alors...

- ...Hors ligne tous les blocs, types de données API, objets technologiques et symboles sont **effacés(!)** et les blocs en ligne, les type de données API, les objets technologiques et les symboles sont chargés dans le projet.
-



7.8. Enoncé : Programmation du mode de fonctionnement et pilotage en mode manuel



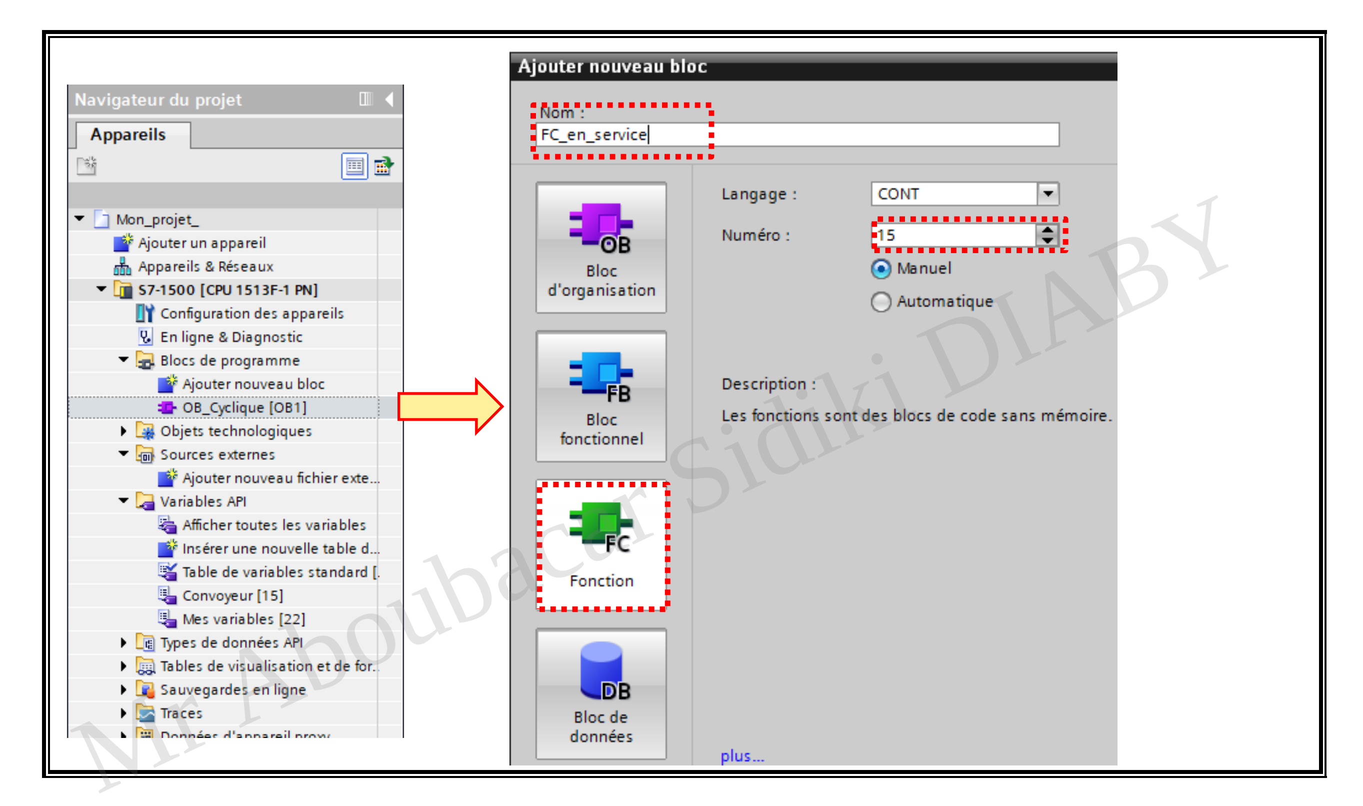
Modes de fonctionnement

Le bouton « BP En Service », permet d'activer le mode « Q En Service ».

Hors Service (« Q En Service »)

Lorsque l'installation est « Hors Service », la bande peut être commandée manuellement vers la droite via le bouton « BP Droite », et vers la gauche via le bouton « BP Gauche ». Si les 2 boutons sont activé appuyés simultanément le convoyeur ne doit pas bouger.

7.8.1. Exercice 1: Ajout du bloc “FC_en_service”



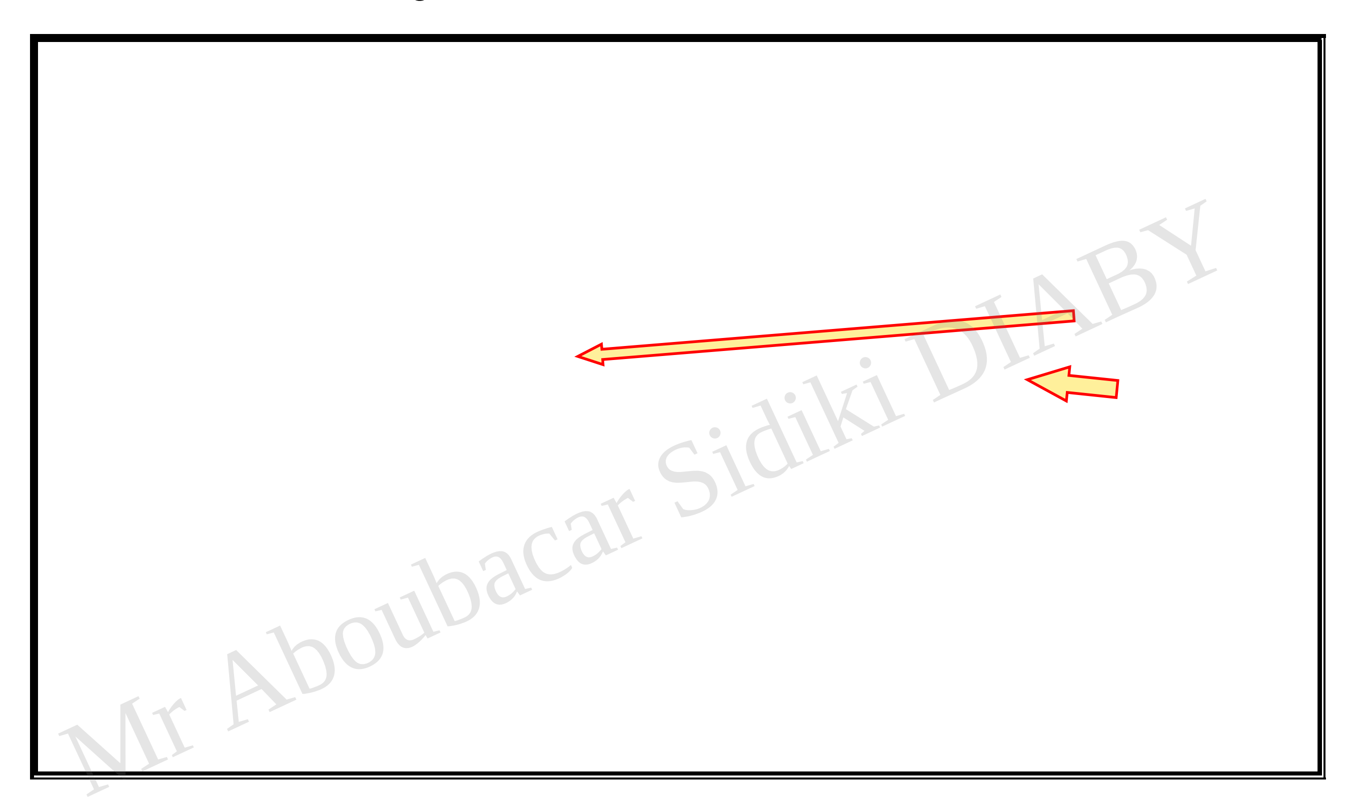
Enoncé

Créez un nouveau bloc « FC_en_service » dans lequel vous allez programmer la marche par impulsions du moteur de la bande transporteuse.

Marche à suivre

1. Double-cliquez dans le dossier « Blocs de programme » sur « Ajouter nouveau bloc ».
2. Paramétrez le nouveau bloc, comme indiqué sur la copie d’écran ci-dessus.

7.8.2. Exercice 2 : Programmation du mode de fonctionnement "FC_En_service



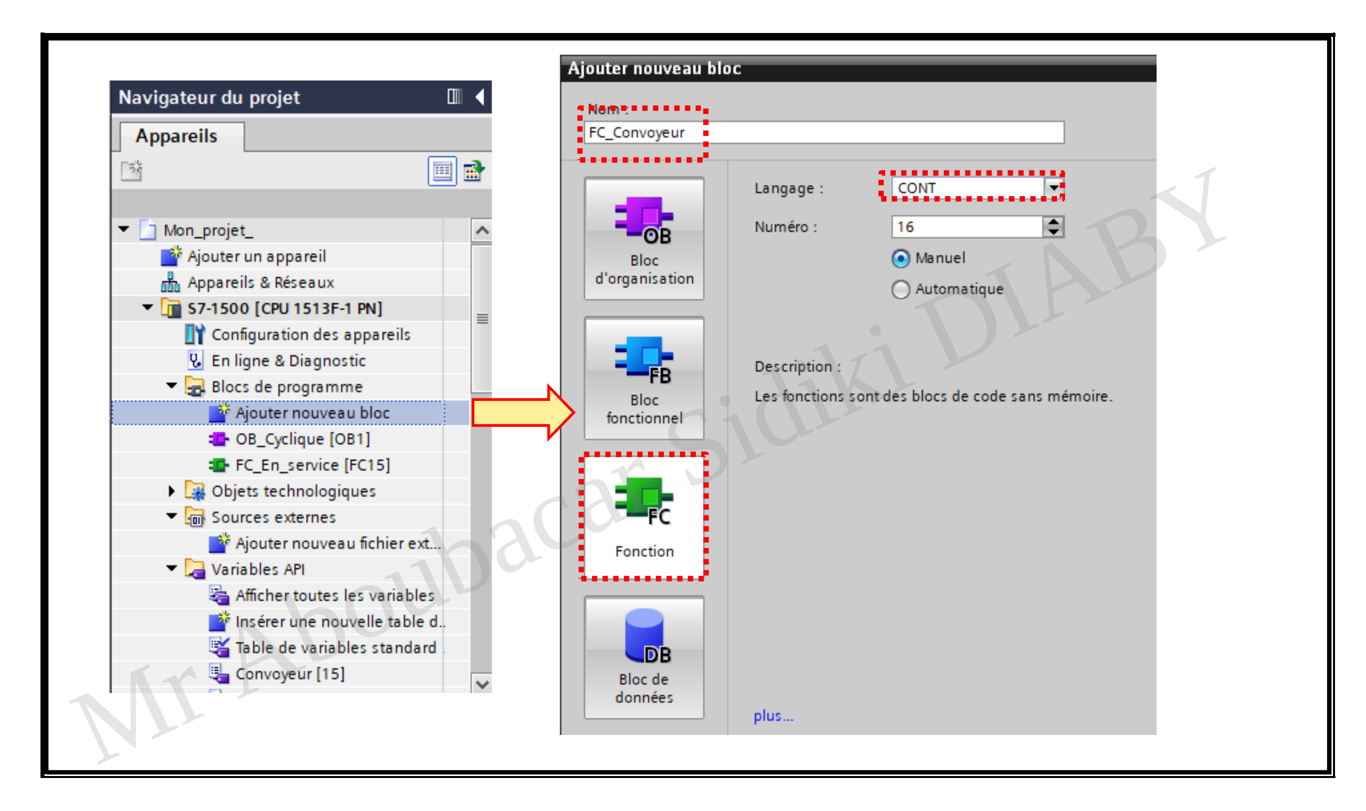
Ennoncé

Programmez le mode de fonctionnement, comme indiqué ci-dessus.

Marche à suivre

1. Programmez une opération par glisser & déposer dans le réseau à partir de la barre des favoris.
2. Sur l'instruction de sortie "Q_En_Service" (Q0.1) (vous pouvez renseigner aussi-bien le symbolique que l'adresse absolue)
3. Affecttion "BP_En_Service" (I 0.1) sur l'instruction d'interrogation.
4. Le commentaire du réseau est automatiquement renseigné à partir du commentaire de la sortie.
5. Fermez le bloc.
6. Sauvegardez le prjet.

7.8.3. Exercice 3: Ajout d'un bloc "FC_Convoyeur"



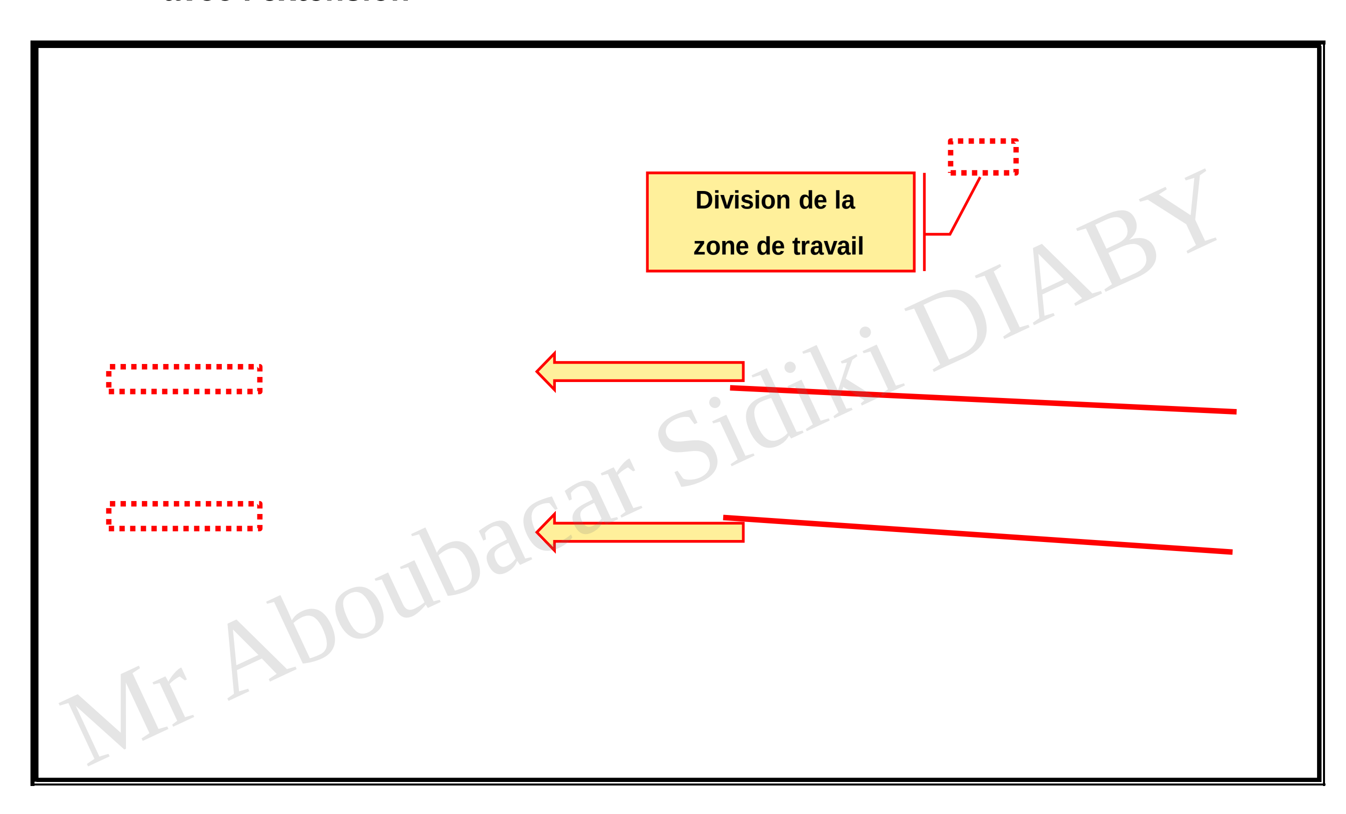
Enoncé

Créez un nouveau bloc « FC_Convoyeur» dans lequel vous allez programmer la marche par impulsions du moteur de la bande transporteuse.

Marche à suivre

1. Double-cliquez dans le dossier « Blocs de programme » sur « Ajouter nouveau bloc ».
2. Paramétrez le nouveau bloc, comme indiqué sur la copie d'écran ci-dessus.

7.8.4. Exercice 4 : Déplacement des réseaux "OB_Cyclique" au "FC_Convoyeur" avec l'extension



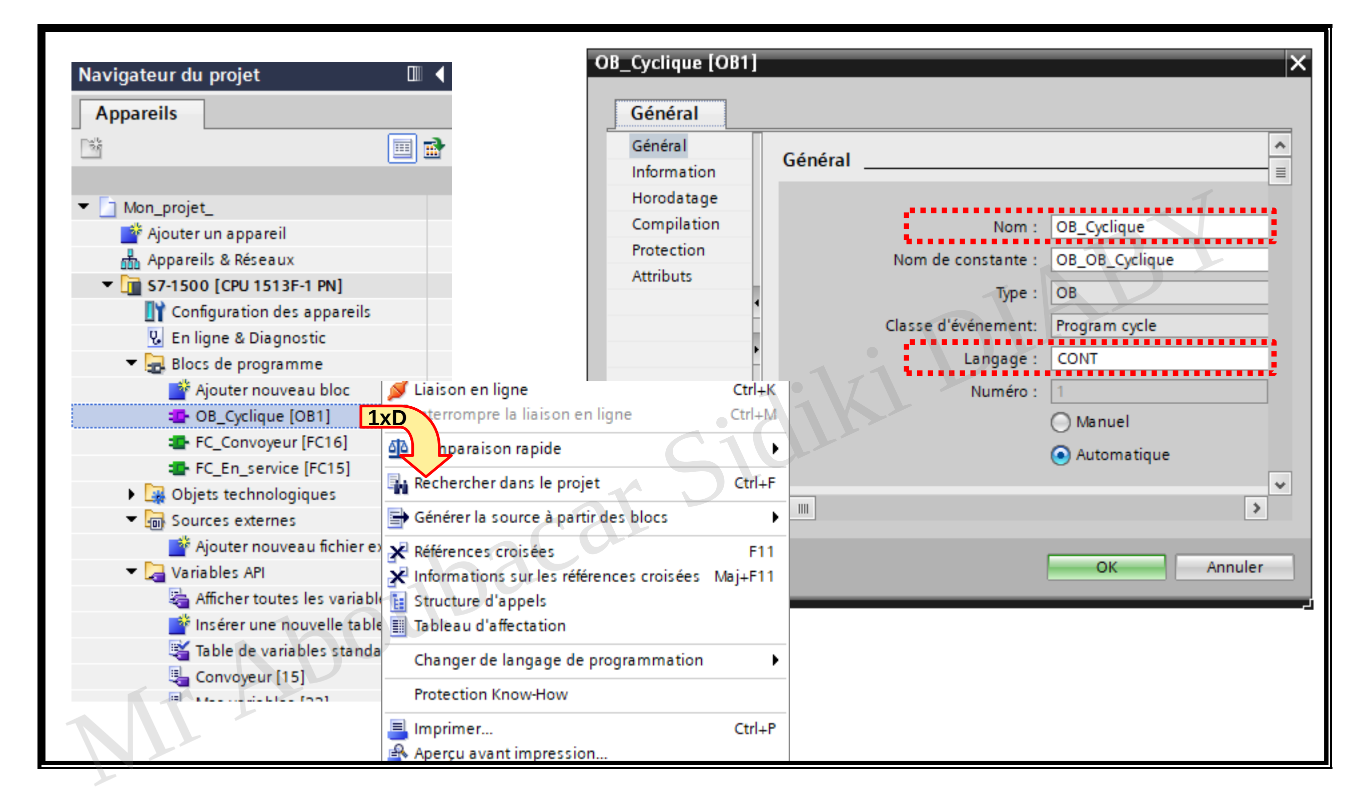
Ennoncé

Programmez le mode pas à pas, comme indiqué ci-dessus.

Marche à suivre

1. Divisez l'écran en 2 (zone de travail) horizontalement ou verticalement avec les bouton appropriés de la barre des taches.
2. Dans chaque zone,ouvrez les blocs "OB_Cyclique" et "FC_Convoyeur".
3. Copiez les réeaux de "OB_Cycle" dans le bloc "FC_Convoyeur" en utilisant la fonction drag en drop.
4. Dans « FC_Convoyeur », reliez la LED « Mode_fct » (Q0.1). du simulateur de manière à ce que le moteur de la bande ne puisse être commandé par impulsions que lorsque celle-ci est éteinte (mode de fonctionnement manuel).
5. Supprimez les 2 réseaux concernés dans "OB_Cyclique".
6. Fermez le bloc.
7. Sauvegardez le prjet.

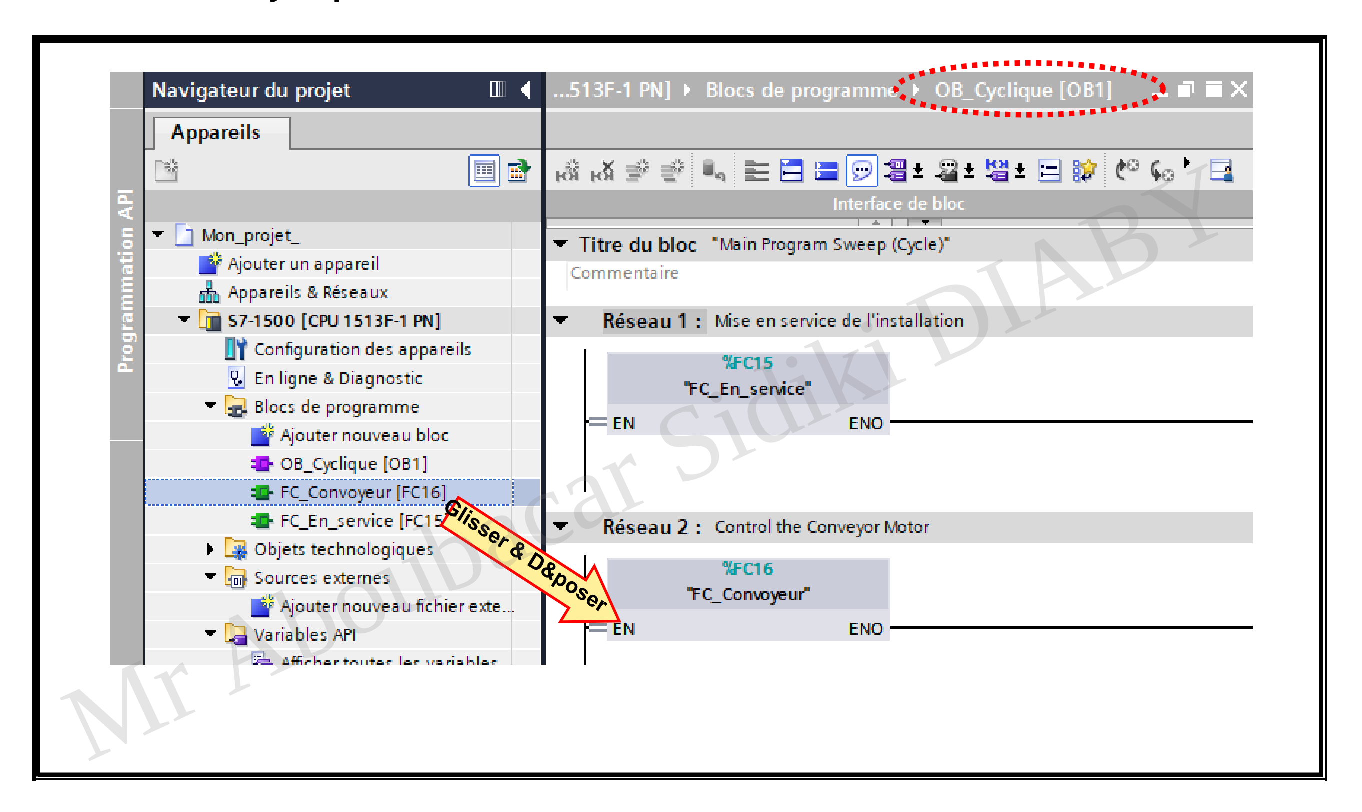
7.8.5. Exercice 5: Vérification de "OB_Cyclique" Propriétés



Ennoncé

Vous pouvez visualiser les prpriétés du bloc comme le montre la figure ci-dessus et les modifier le cas échéant.

7.8.6. Exercice 6: Appel des blocs "FC_En_service" et "FC_Convoyeur" dans "OB_Cyclique"



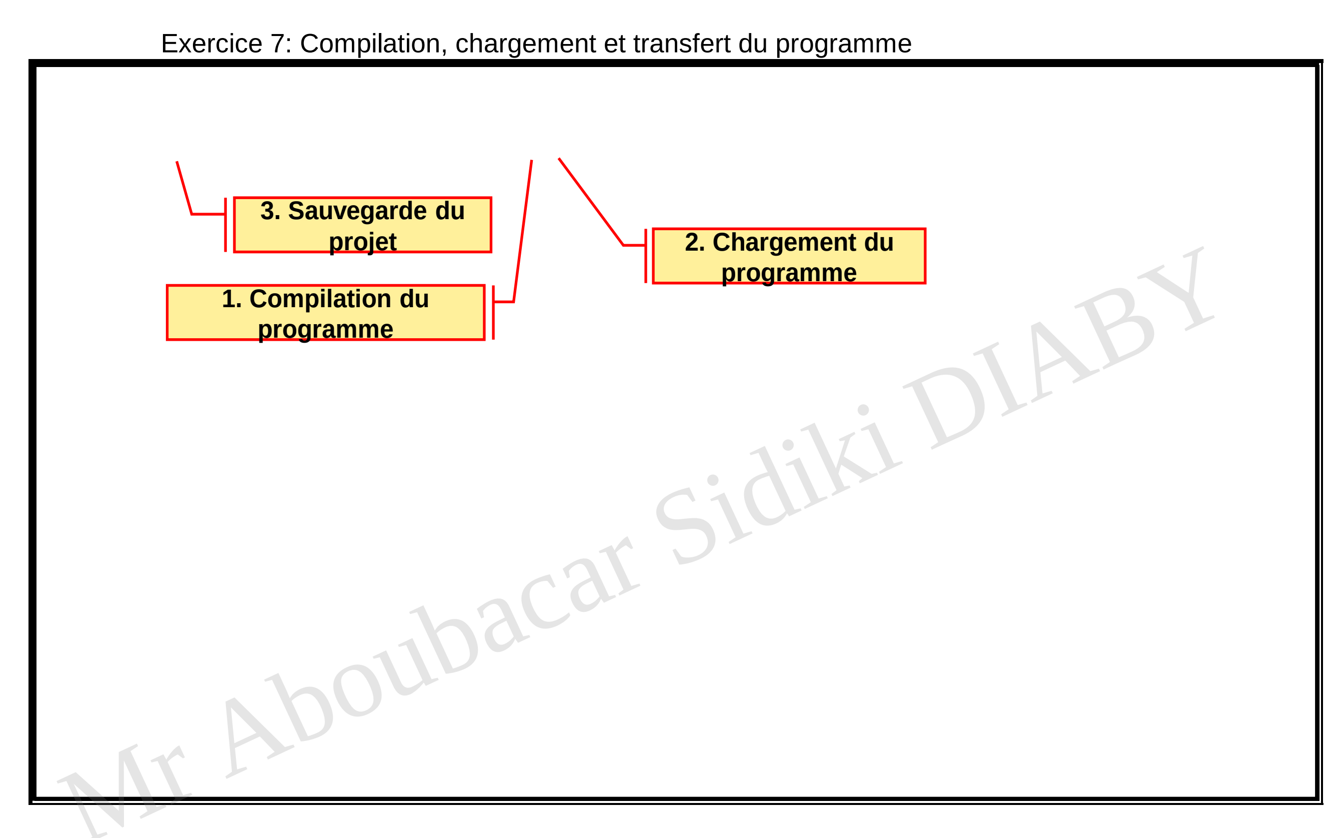
Ennoncé

Pour assurer le traitement cyclique des blocs nouvellement créés «, il faut programmer un appel des blocs dans l'OB Main.

Marche à suivre

1. Ouvrez le bloc « OB_Cyclique » par un double-clic.
2. Programmez l'appel des blocs « FC_En_Service » et « FC_Convoyeur » par glisser & déposer, comme indiqué ci-dessus.
3. Editez les titres de réseau.
4. Fermez les blocs.
5. Sauvegardez le prjet.

Exercice 7: Compilation, chargement et transfert du programme



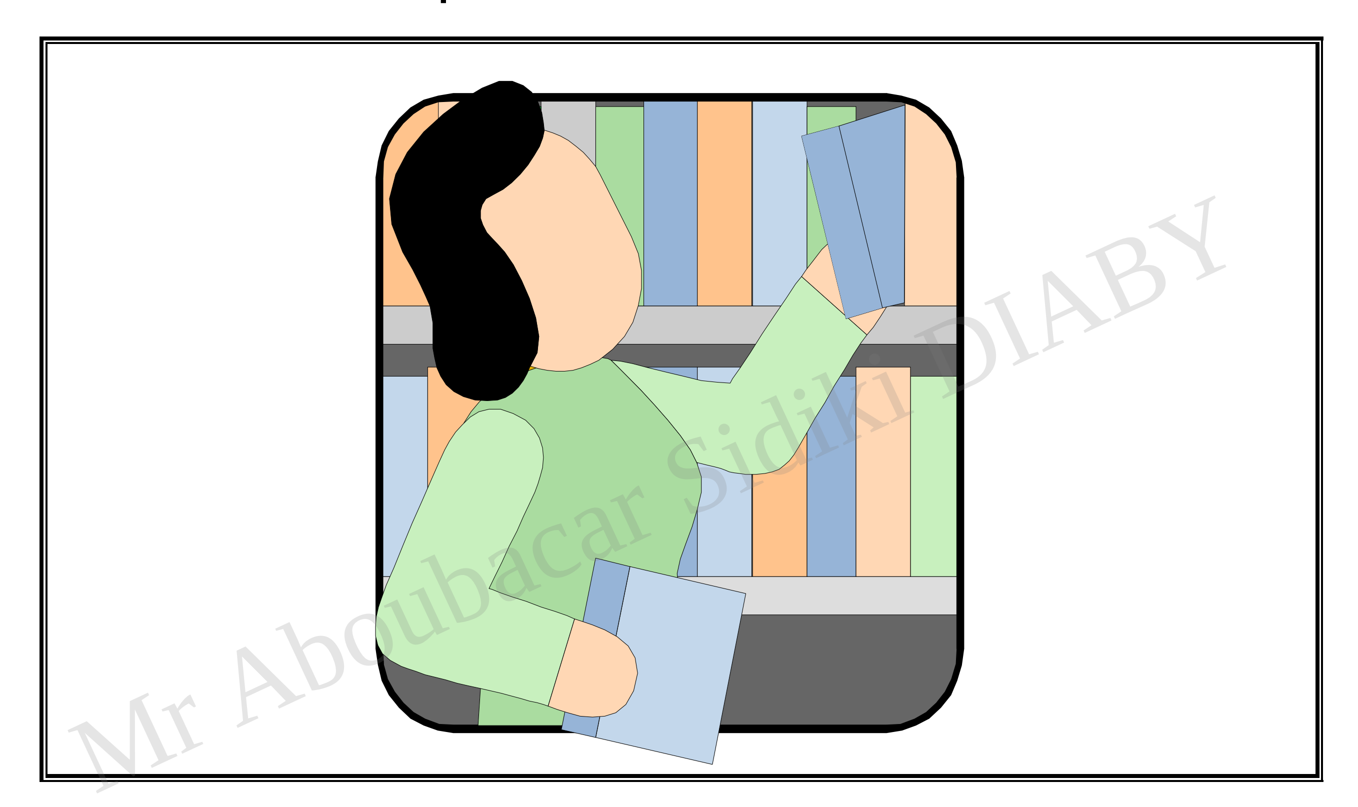
Enoncé

Les blocs nouvellement programmés doivent être compilés, chargés dans la CPU et enregistrés hors ligne dans la gestion des données du projet.

Marche à suivre

1. Pour compiler le programme complet et charger dans la CPU, sélectionnez le dossier « Blocs de programme » dans le navigateur de projet.
2. Effectuez les opérations décrites ci-dessus et contrôlez le fonctionnement du programme en appuyant sur les boutons du simulateur « BP_droite » (I0.2) et « BP_gauche » (I0.3) en choisissant le bon mode de fonctionnement avec « BP_En_service »

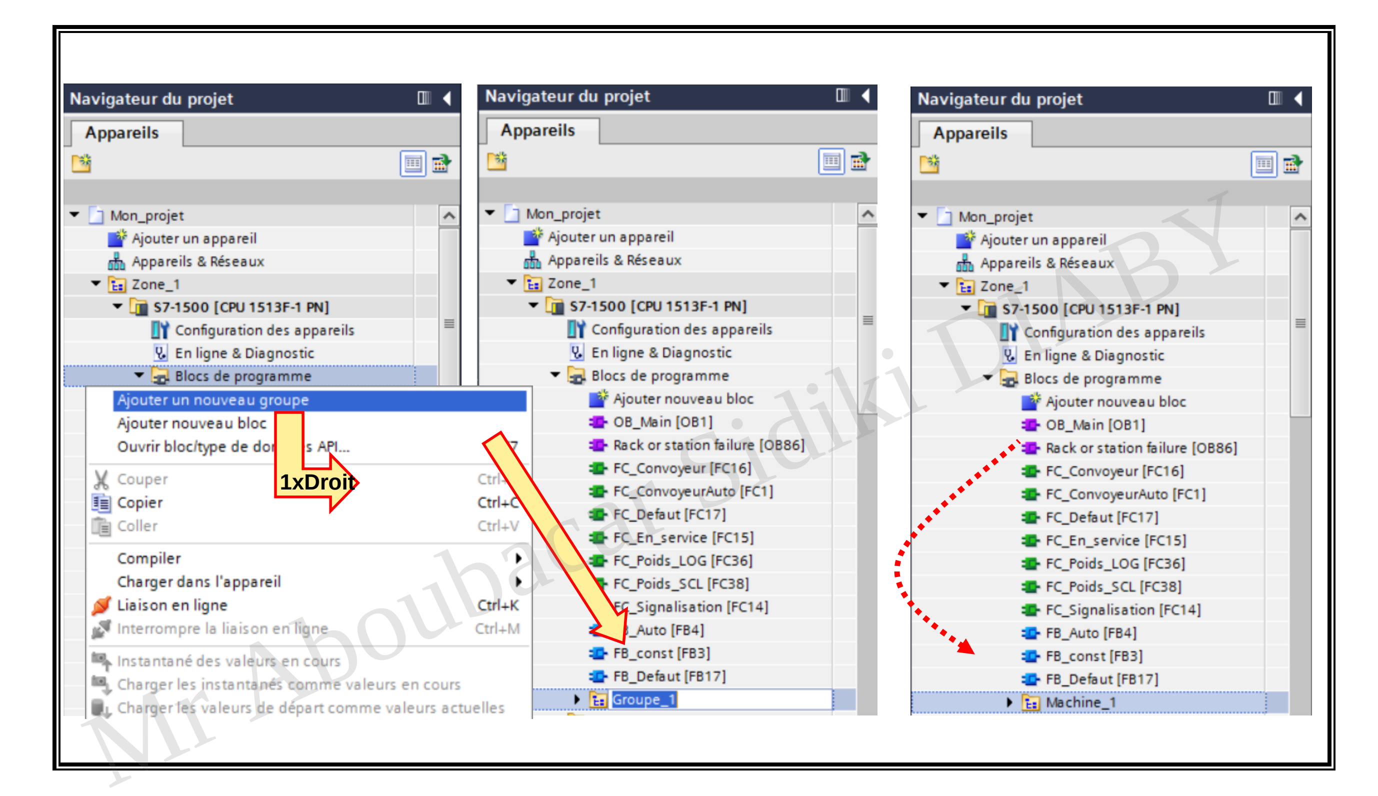
7.9. Pour en savoir plus…





7.9.1. Groupes de blocs





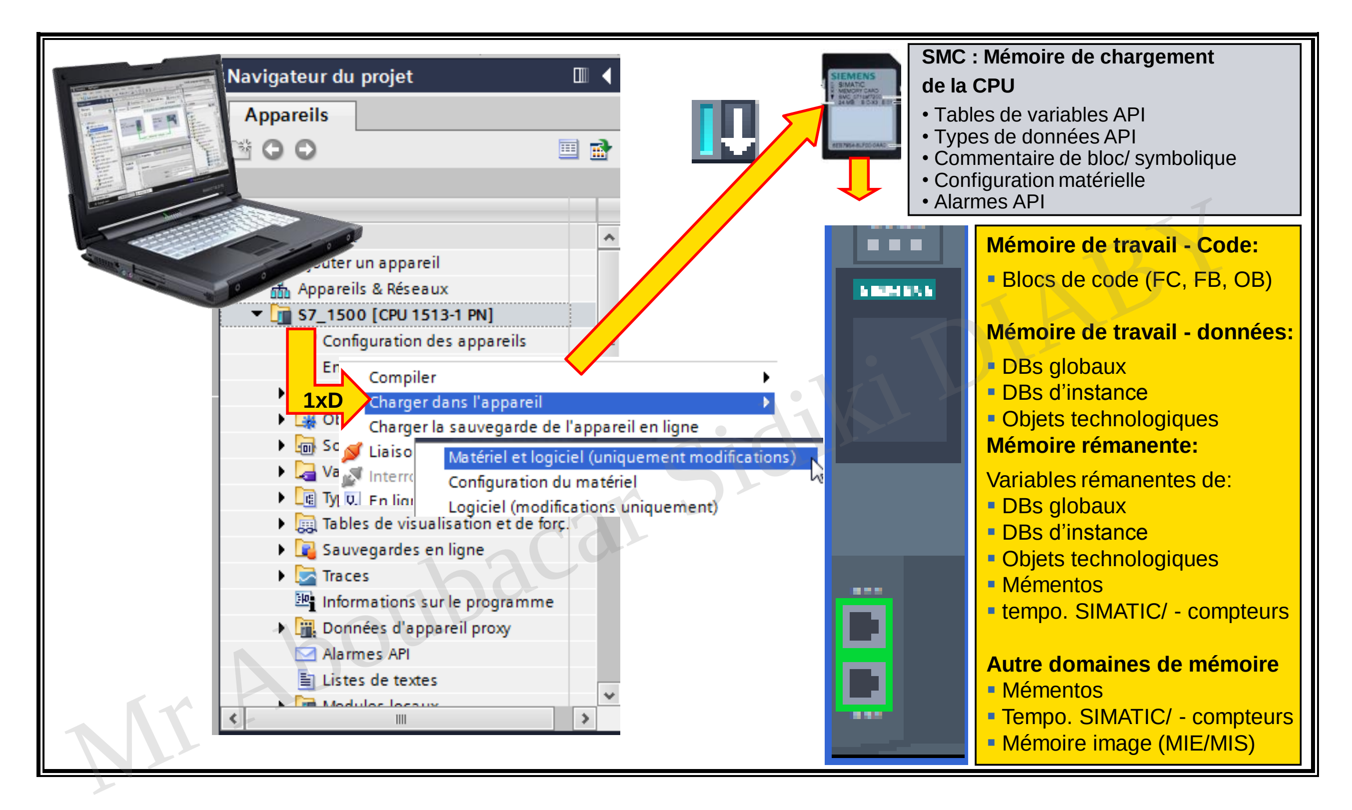
Groupes de blocs

Pour garantir une meilleure lisibilité, les programmes de grande envergure composés de nombreux blocs peuvent être subdivisés en différents groupes de blocs. Les regroupements peuvent être réalisés en fonction de la structure de l'installation à commander.

Même si les blocs sont gérés dans différents groupes, chaque bloc doit être doté d'un nom symbolique unique. La somme de tous les blocs représente le programme utilisateur, indépendamment des regroupements.

Le déplacement de blocs entre les différents groupes s'effectue par simple glisser-déposer.

7.9.2. Chargement des blocs dans la CPU: S7-1500 – Système de mémoire



Zones mémoire de la CPU

La carte SD est la mémoire de chargement de la CPU. Cela implique qu'une carte insérée est absolument nécessaire au fonctionnement de la CPU.

La carte SD contient les données de la station complète c'est à dire le programme S7 complet avec la documentation, les variables APi et les types de données API ainsi que la configuration matérielle avec la périphérie décentralisée et le paramétrage. De plus, la carte SD peut encore contenir d'autres données comme par ex. des recettes , des sauvegardes HMI.....

Lors du chargement d'un programme S7 dans la CPU, tous les blocs sont tout d'abord chargés dans la mémoire de chargement puis, à partir de là, la CPU copie automatiquement les éléments nécessaires au traitement des blocs dans la mémoire de travail.

Mémoire de travail

La mémoire de travail est une mémoire RAM volatile qui n'est pas extensible.Elles se divise en deux parties:

- La mémoire de travail du code qui contient les éléments nécessaires à l'exécution des blocs de code
- La mémoire de travail des données qui contient les éléments nécessaires au traitement des blocs de données et des objets technologiques.

Mémoire rémanente dépend de la CPU (88-700 kOctets)

La mémoire rémanente est une mémoire non volatile servant à la sauvegarde de l'ensemble des variables déclarées rémanentes dont les valeurs restent conservées en cas de coupure de tension.

Le contenu de la mémoire rémanente est uniquement effacé par

- l'effacement général
- la restauration des paramètres d'usine de la CPU

Sommaire

8

8. Opérations binaires

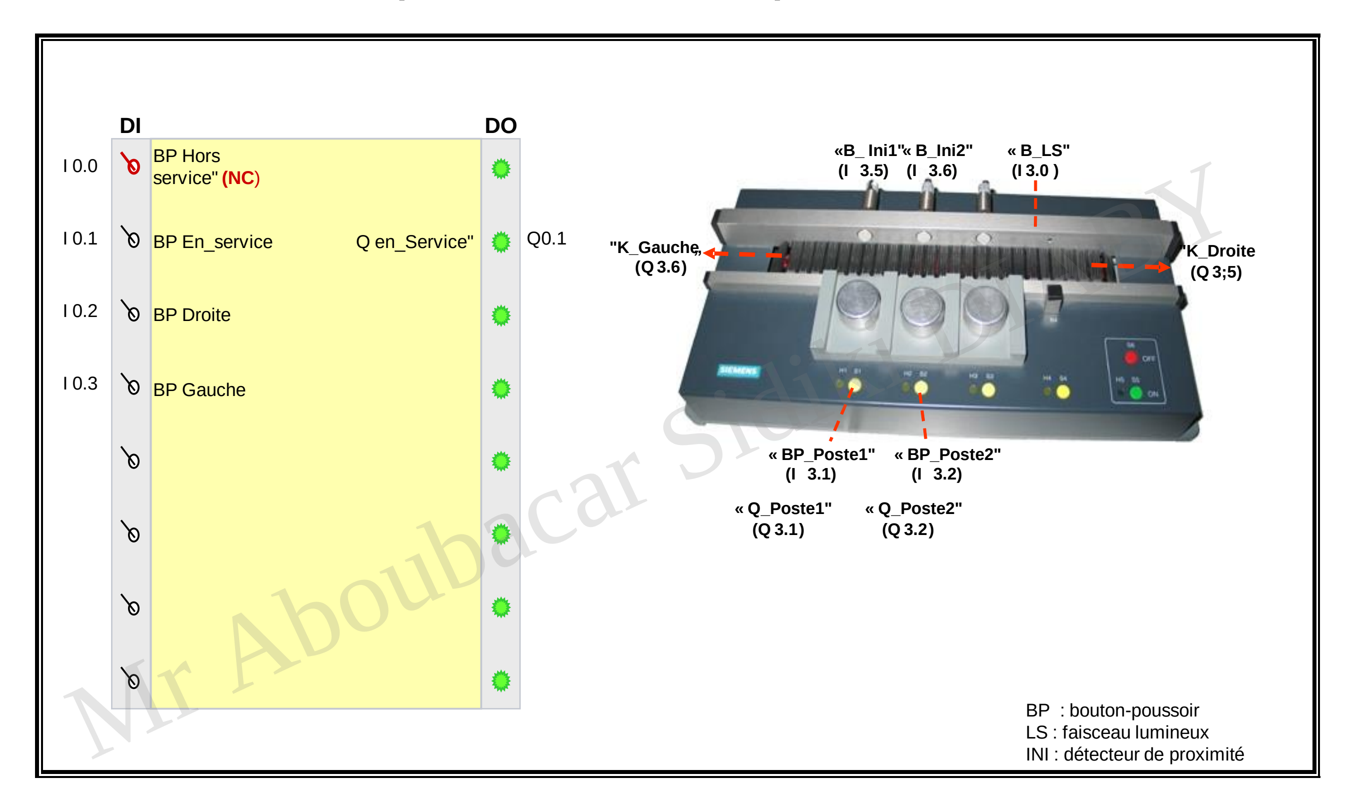
A l'issue du chapitre, vous allez …

… être familiarisé avec les instructions de Set, Reset

… savoir gérer les fronts montants

… être familiarisé avec les opérations de saut

8.1. Enoncé : Maquette de bande transporteuse



Maquette de bande transporteuse

La bande transporteuse sert à transporter des pièces d'un poste 1 ou 2 jusqu'à un poste barrière lumineuse. Les boutons du simulateur « BP_marche » (I0.1) et « BP_arret » (I0.0, NF) permettent de mettre en route et d'arrêter l'installation (LED du simulateur « Q_en_service », Q0.1).

- Lorsque « Q_en_service » (Q0.1) est désactivé...

le moteur de la bande peut être commandé par impulsions vers la droite et vers la gauche via les boutons du simulateur « BP_droite » (I0.2) et « BP_gauche » (I0.3).

- Lorsque « Q_en_service » (Q0.1) est activé...

...les pièces sont transportées du poste 1 ou 2 jusqu'à la barrière lumineuse. Les pièces ne doivent être déposées qu'à un seul des deux postes et le bouton correspondant doit être actionné.

Les LED des postes 1 et 2...

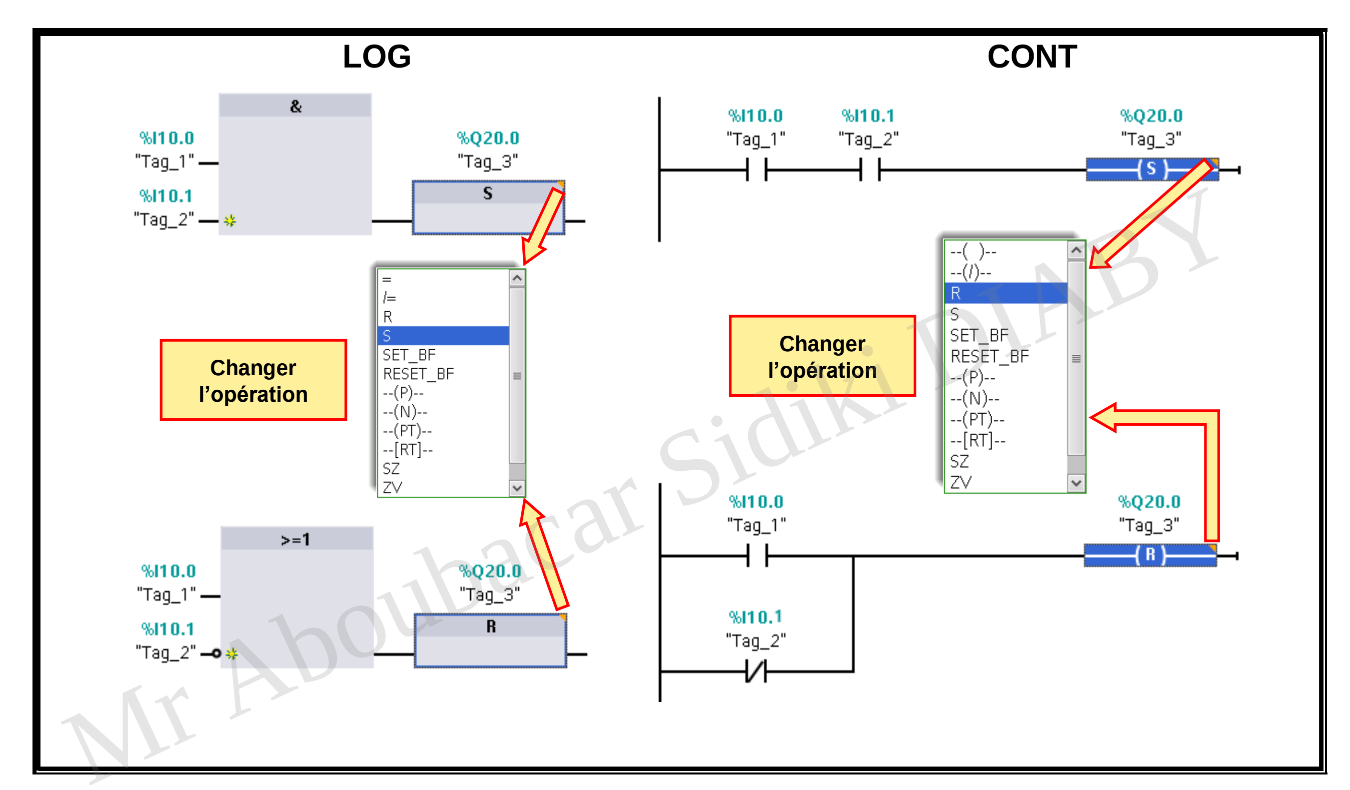
...sont allumées en fixe lorsqu'une nouvelle pièce peut être déposée sur la bande (le moteur de la bande est à l'arrêt et les deux détecteurs sont libres)

...clignotent à 1 Hz au poste où une pièce a été détectée par le détecteur correspondant tant que la bande n'est pas activée (si des pièces sont déposées devant les deux détecteurs, aucune LED ne doit s'allumer)

... clignotent à 2 Hz pendant toute la durée de fonctionnement du moteur de la bande.

La LED du poste barrière lumineuse clignote à 2 Hz lorsque le moteur de la bande fonctionne et passe en allumage fixe lorsque la consigne de pièces à transporter est atteinte.

8.2. Affectation, mise à 1, mise à 0



l'affectation et peut être affecté à un autre opérande ou relié à d'autres opérations.

Mise à 1

Si RLG = « 1 », l'opérande indiqué est mis à 1, si RLG = « 0 », l'état de l'opérande reste inchangé.

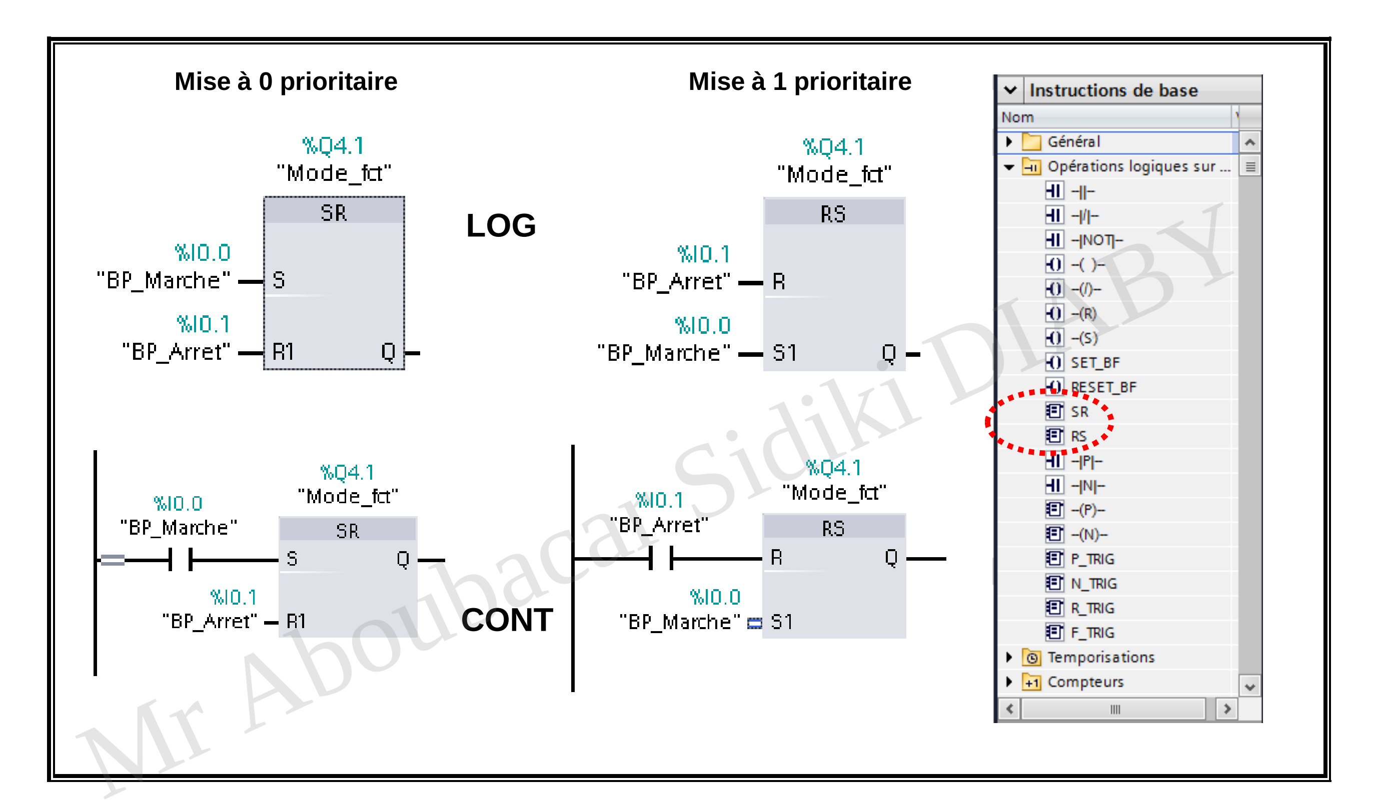
Mise à 0

Si RLG = « 1 », l'opérande indiqué est mis à 0, si RLG = « 0 », l'état de l'opérande reste inchangé.

Note

En addition de la fonction SET et RESET, le graphique ci-dessus montre la méthodologie pour changer l'attribut des éléments sans devoir supprimer l'élément existant et insérer un nouveau type d'élément. Sélectionner l'élément et cliquer sur la petite « flèche » rouge situé en haut à droite de l'élément. Les changements disponibles seront montrés dans un menu déroulant.

8.2.1. Bascules mise à 1 / mise à 0



Bascule

Une bascule possède une entrée de mise à 1 et une entrée de mise à 0. Selon l'entrée à laquelle un RLG = « 1 » est présent, l'opérande est mis à 1 ou à 0.

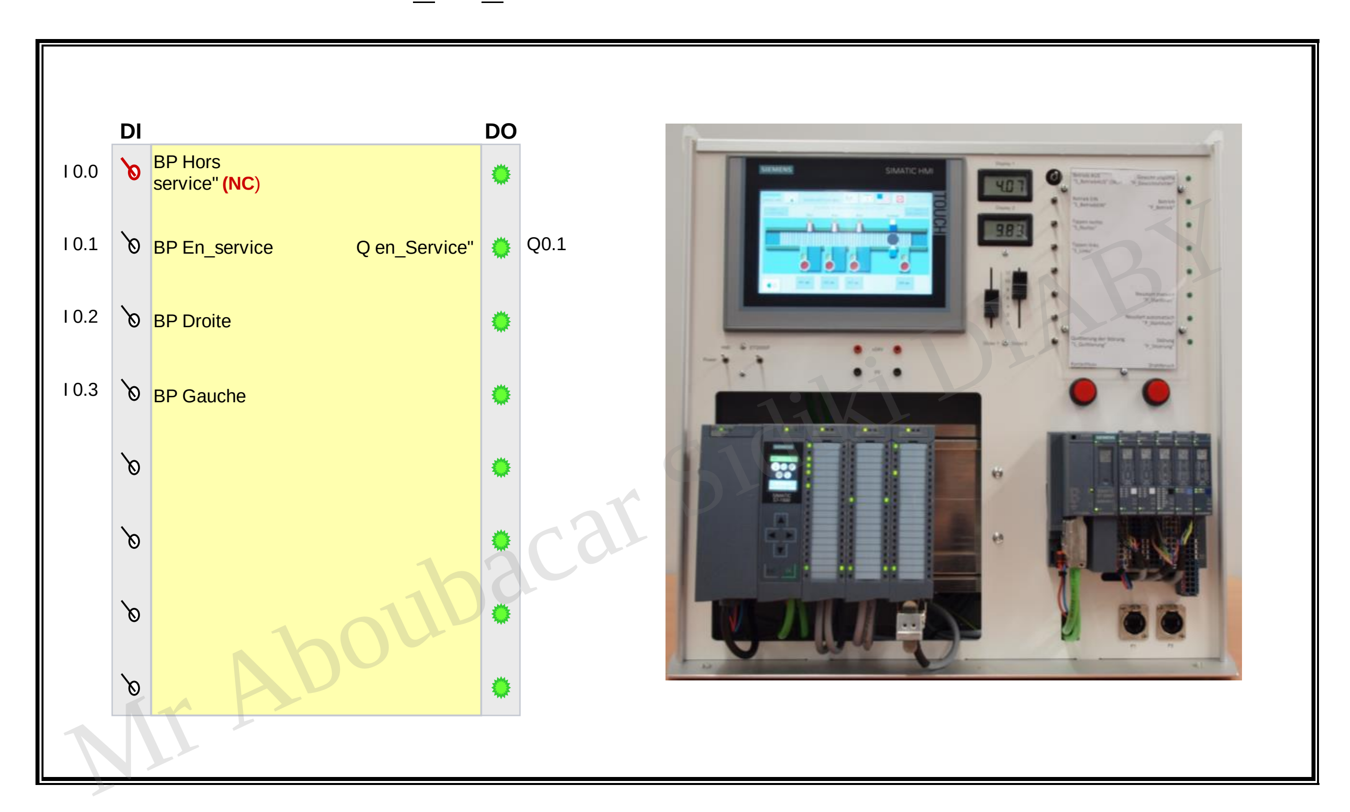
Priorité

Si les deux entrées présentent simultanément un état de signal ou un RLG = « 1 », la priorité de l'opération de mise à 1 ou de mise à 0 sera déterminante. En CONT et LOG, on dispose de ce fait de symboles différents pour la mise à 1 prioritaire et pour la mise à 0 prioritaire.

Remarque

En cas de redémarrage de la CPU, toutes les sorties sont remises à zéro ou leur état écrasé par un « 0 ».

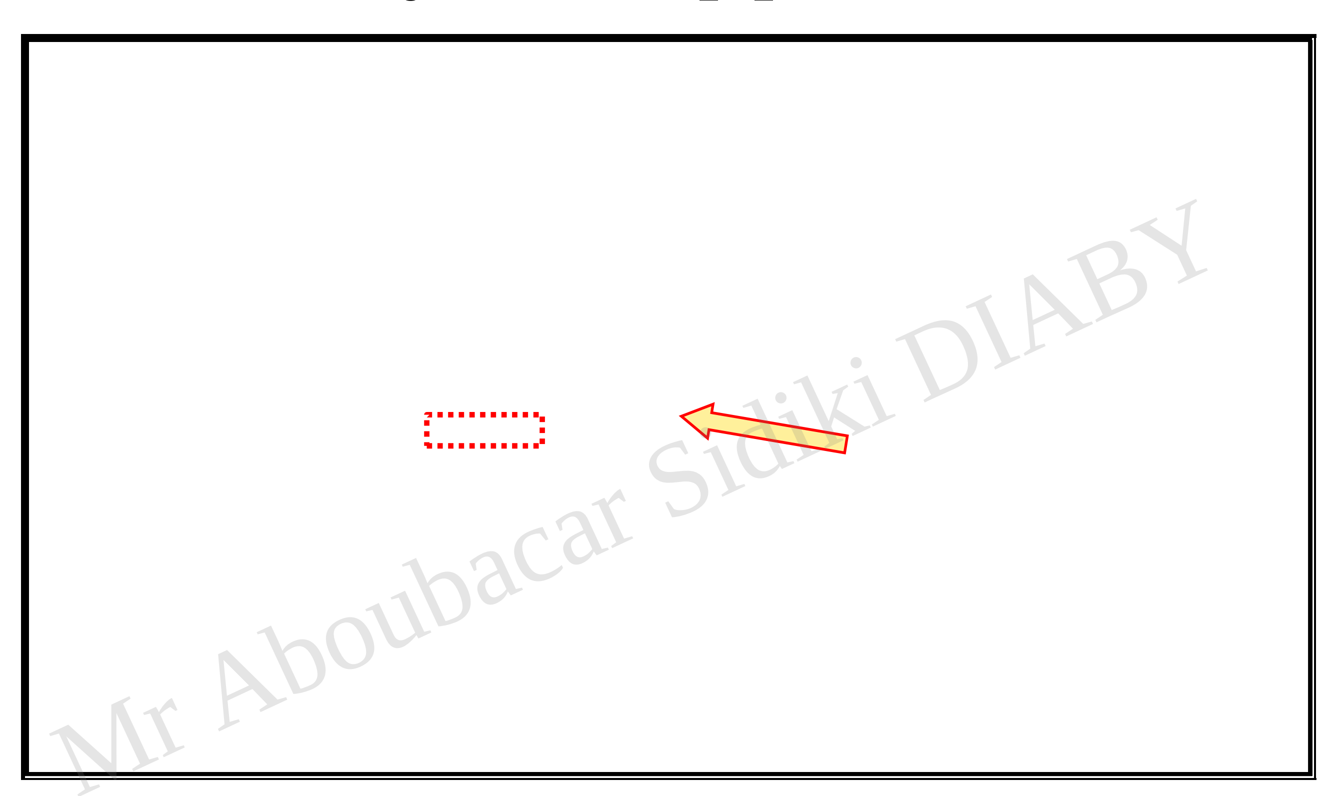
8.3. Enoncé : « FC_En_Service »



Enoncé

- Mode de fonctionnement dans « FC_En_service » :
  Le bouton du simulateur « BP_en_service» permet de démarrer l'installation ou d'activer « Q_en_service » » (Q0.1), le bouton du simulateur « BP_Hors_Service » (NC) de mettre l'installation à l'arrêt.

8.3.1. Exercice 1 : Programmation du “FC_en_service”



Enoncé

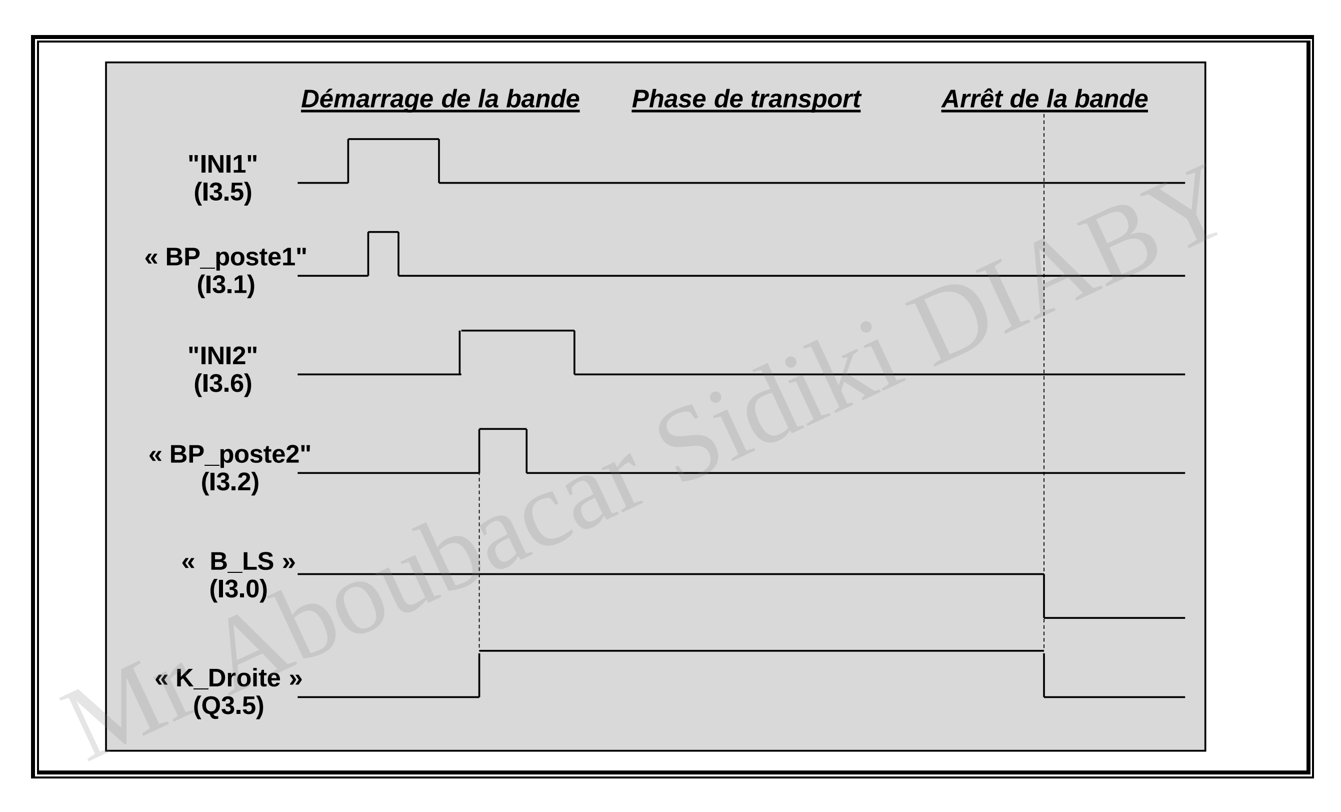
Modifiez le programme « FC_en_service » une séquence de commande de la bande transporteuse :

Le bouton du simulateur « BP_en_service » (I0.1) permet de démarrer l'installation ou d'activer la LED du simulateur «Q_en_service» (Q0.1), le bouton « BP_hors_service » (I0.0) de mettre l'installation à l'arrêt.

Marche à suivre

1. Ouvrez le bloc « FC_Mode_fonctionnement » .
2. Supprimez le programme réalisé dans le chapitre précédent
2. Ouvrez l'onglet « Instructions », programmez par glisser-déposer les instructions indiquées dans la figure ci-dessus en utilisant les adresses absolues indiquées.
3. Renommez les variables utilisées comme indiqué dans la figure ci-dessus.
4. Appelez le bloc « FC_Mode_fonctionnement » dans l'OB Main.
4. Dans « FC_Convoyeur » , reliez la LED « Q_en_service » (Q0.1). du simulateur de manière à ce que le moteur de la bande ne puisse être commandé par impulsions que lorsque celle-ci est éteinte (mode de fonctionnement manuel).
5. Chargez les blocs modifiés dans la CPU et vérifiez le fonctionnement du programme.

8.3.2. Enoncé : Transport des pièces lorsque la sortie « Q_en_service » (Q4.1) est activée



Enoncé

Lorsque « Q_en_service » (Q0.1) est activé, les pièces doivent être transportées du poste 1 ou 2 vers la barrière lumineuse « B_LS » (I3.0). Les pièces à transporter ne doivent être déposées qu'à un seul des deux postes de la bande. Si des pièces sont déposées aux deux postes, l'opération de transport ne peut pas être lancée.

- Le moteur de la bande « K_Droite » (Q3.5) démarre si
  - le détecteur « Ini1_Detecteur_poste1» (I3.5) du poste 1 est occupé ET NON le détecteur « Ini2_Detecteur_poste2» (I3.6) du poste 2 ET le bouton du poste 1 « BP_Poste1 » (I3.1) est actionné

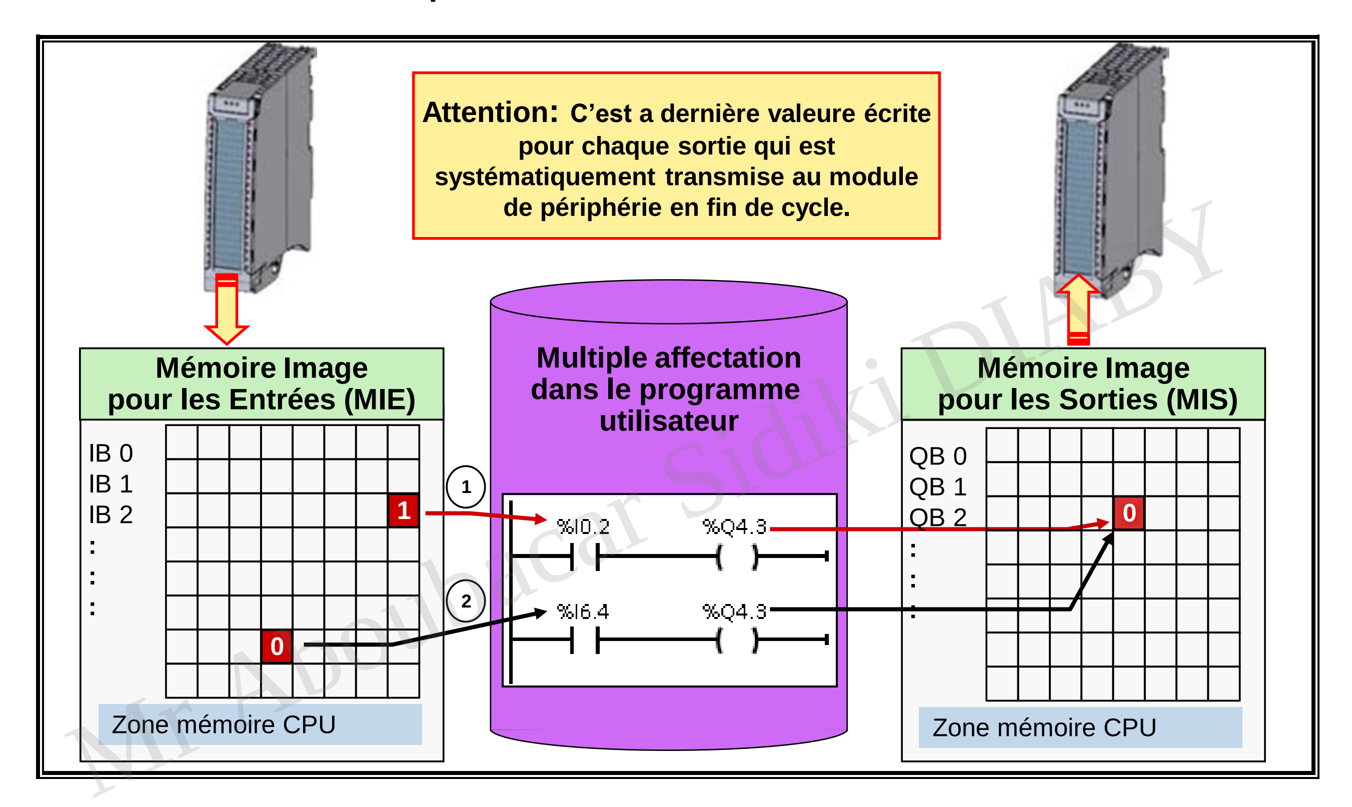
  OU

  - le détecteur « Ini2_Detecteur_poste2» (I3.6) du poste 2 est occupé ET NON le détecteur « Ini1_Detecteur_poste1» (I3.5) ET le bouton du poste 2 « BP_Poste2» (I3.2) est actionné.
- Le moteur de la bande « K_Droite » (Q3.5) s'arrête si
  - la pièce a atteint la barrière lumineuse « B_LS » (I3.0)

  OU

  - « Q_en_service » (Q0.1) est désactivé.

8.3.3. Affectation multiple de sorties



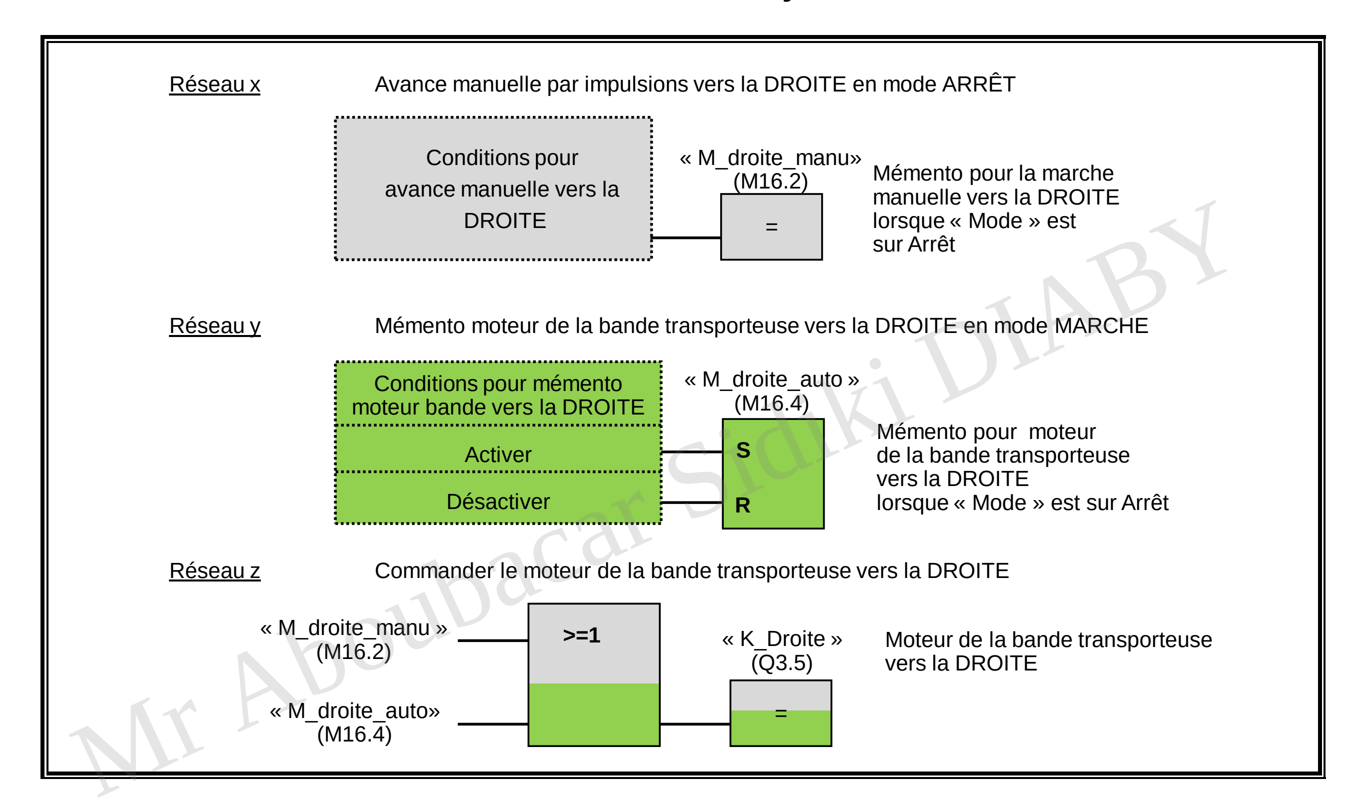
Mémoires image du processus

Pour stocker les états logiques des entrées et sorties TOR interrogées, la CPU dispose de zones mémoire réservées : la mémoire image des entrées (MIE) et la mémoire image des sorties (MIS). Lors du traitement, le programme utilisateur accède à ces zones mémoire, et non pas directement aux modules d'entrée et sortie TOR.

"Erreur Classique": Double Assignation

Si une sortie est assignée dans plusieurs parties du programme, alors seulement le dernier statut qui a est assigné sera transférer au module de sortie. Ce type de double assignation est considérer comme des erreurs de programmation.

8.3.4. Exercice 3 : Etendre le bloc « FC_Convoyeur »



Enoncé

Etendez le bloc « FC_Convoyeur » avec les fonctions décrites ci-dessus.

Aide à la programmation

La commande du moteur de la bande « K_Droite » (Q3.5) doit à présent répondre à deux conditions :

Commande par impulsions vers la DROITE avec « Q_en_service » (Q0.1) désactivé OU commande dans les conditions décrites plus haut (réseau y dans la figure ci-dessus) avec « Q_en_service » (Q0.1) activé.

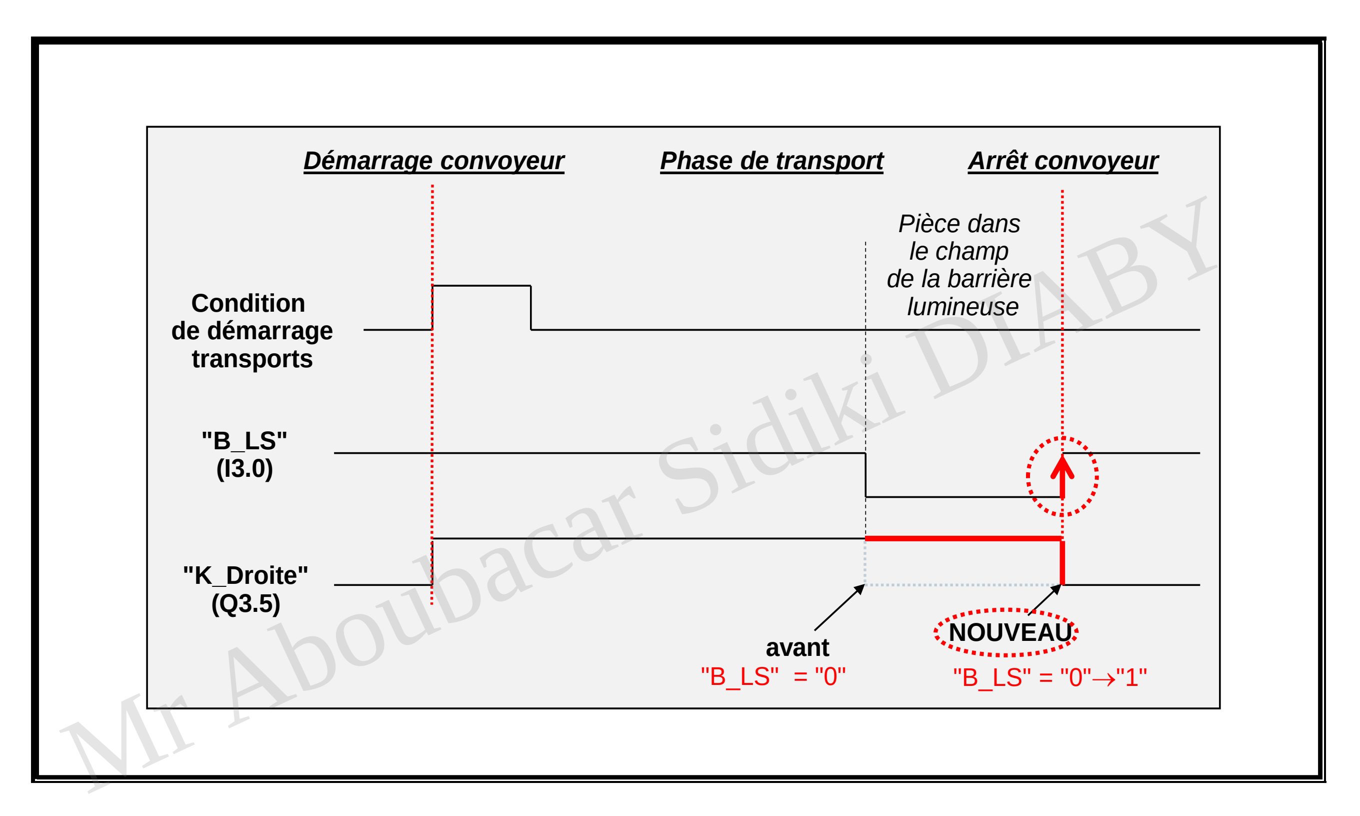
Si le résultat logique des conditions avait été directement affecté à la sortie « K_Droite » (Q3.5) dans les deux réseaux x et y, une erreur serait apparue en raison d'une double affectation. La commande par impulsions vers la DROITE du moteur en mode manuel (réseau x) ne fonctionnerait plus, car l'état affecté ici à la sortie serait ensuite écrasé dans le réseau y.

Pour résoudre ce problème, il suffit de programmer un mémento pour chacune des deux conditions ou d'affecter dans les deux réseaux x et y chaque résultat des opérations à un mémento. Ces mémentos serviront ensuite à la commande du moteur de la bande dans le réseau z.

Marche à suivre

1. Ouvrez le bloc « FC_Convoyeur » .
2. Programmez la fonction souhaitée dans de nouveaux réseaux (voir figure). Utilisez pour ce faire les mémentos indiqués dans la figure et affectez-leur les symboles indiqués.
3. Modifiez le réseau « Marche droite » existant comme indiqué dans l'aide à la programmation.
4. Chargez le bloc étendu dans la CPU et vérifiez le fonctionnement correct du programme.

8.4. Enoncé : Franchissement de la barrière lumineuse



Fonction précédente dans FC16

Lorsque « Q_en_service » est désactivé (Q0.1 = « 0 »), le moteur de la bande « K_Droite » (Q3.5) peut être commandé par impulsions à l'aide des boutons du simulateur « BP_gauche » (I0.3) et « BP_droite » (I0.2).

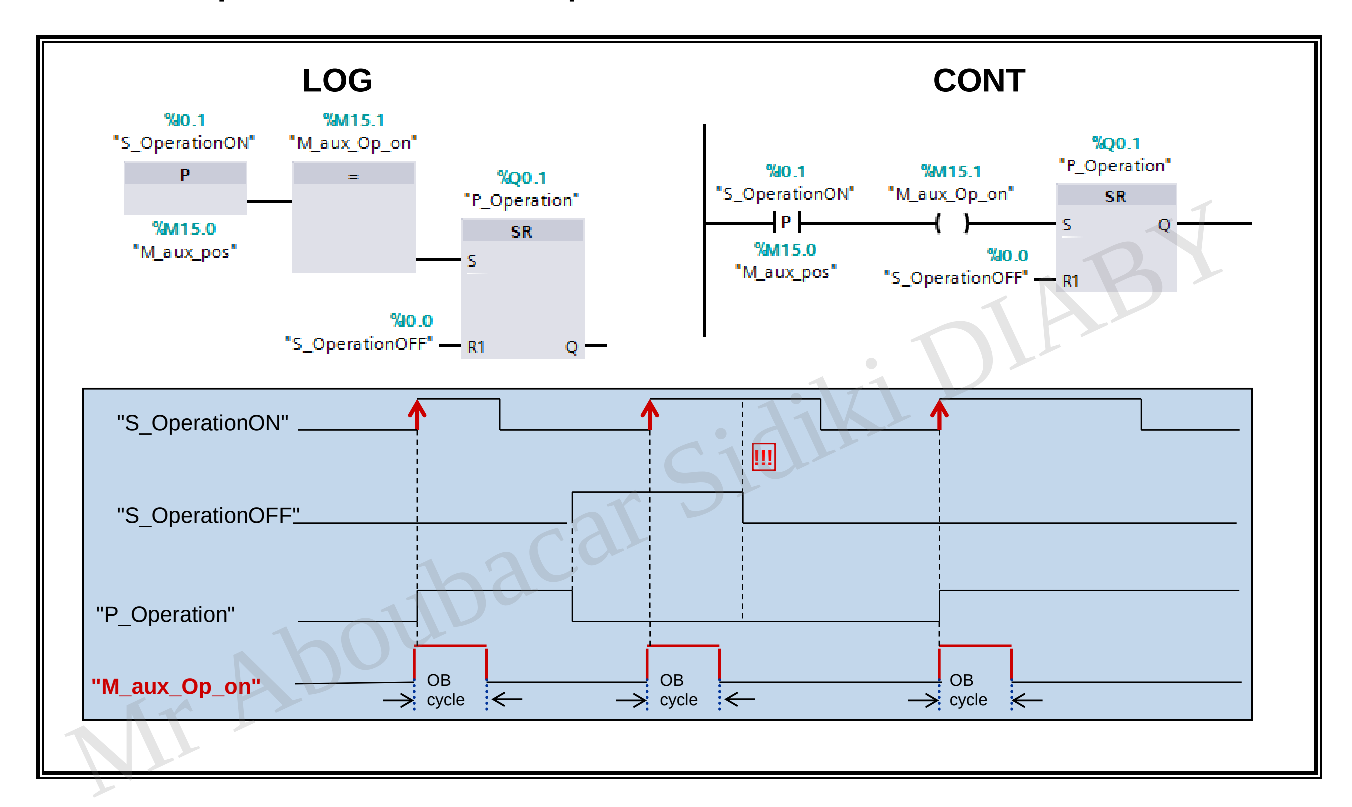
Lorsque « Q_en_service » (Q0.1) est activé, le moteur de la bande « K_Droite » (Q3.5) est activé lorsqu'une pièce est déposée devant le détecteur de proximité d'un seul des postes 1 ou 2 et que le bouton du poste occupé est actionné.

Le moteur « K_Droite » (Q3.5) s'arrête lorsque la pièce a **atteint** la barrière lumineuse ou que « Q_en_service » (Q0.1) est désactivé.

Enoncé

La fonction du bloc « FC_Convoyeur » pour la commande du moteur lorsque « Q_en_service » (Q0.1) est activé reste inchangée dans le principe, mais le moteur ne doit s'arrêter que lorsque la pièce a **franchi** la barrière lumineuse.

8.4.1. Réponse aux fronts des opérandes



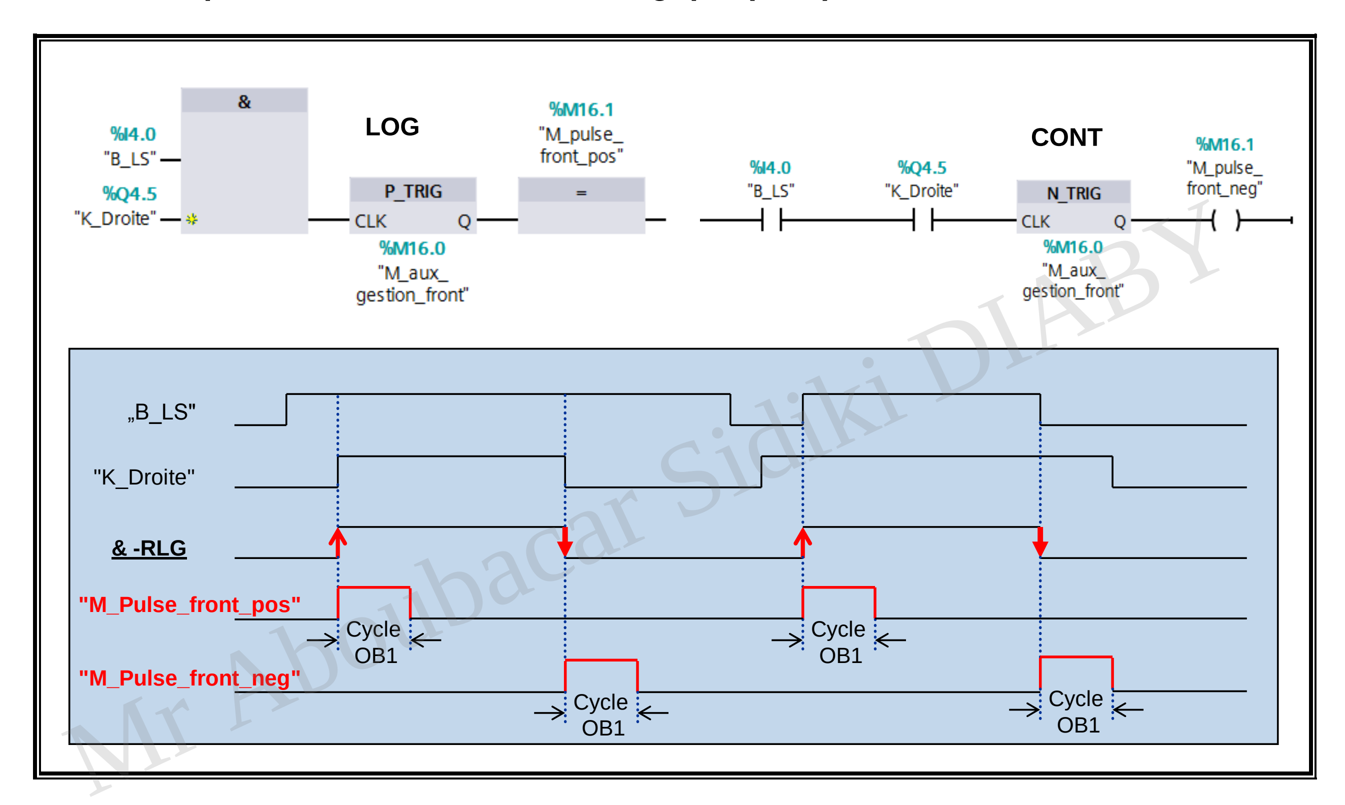
Evaluation du front d'un opérande (P, N)

L'évaluation du front d'un opérande permet de détecter le passage de l'état « 0 » à « 1 » (front montant ou positif) ou de « 1 » à « 0 » (front descendant ou négatif) d'un opérande (« S_OperationON » dans l'exemple). Dans ce cas, l'instruction fournit le résultat logique RLG « 1 », ce dernier peut être utilisé par la suite (dans l'exemple en tant que condition de mise à 1 mémorisée) ou affecté comme état à un autre opérande (par ex. un mémento). Lors du cycle suivant, l'instruction fournit à nouveau le résultat « 0 » même si (« S_OperationON » reste à l'état « 1 ».

L'instruction compare l'état actuel de l'opérande (« S_OperationON » à l'état qu'il avait lors du cycle de traitement du programme précédent. Cet état est mémorisé à cet effet dans un mémento dit mémento de front (dans l'exemple « M_aux_Op_on »). Vous devez vous assurer à ce que l'état de ce mémento de front ne soit pas écrasé à un autre endroit du programme. Vous utiliserez pour ce faire un mémento de front distinct, même si le même opérande (dans l'exemple « S_OperationON » doit être réévalué par ex. dans un autre bloc!

De plus dans l'exemple, le front est assigné à un mémento de front « M_aux_Op_on »). Cela présente l'avantage d'une évaluation unique du front de l'opérande dans le programme. S'il faut réévaluer le front de l'opérande dans le traitement ultérieur du programme, alors il ne sera pas nécessaire d'utiliser un autre mémento de front, car l'état de ce mémento de front pourra être interrogé.

8.4.2. Réponse aux fronts – résultat logique (RLG)



Réponse aux fronts (montant ou descendant)

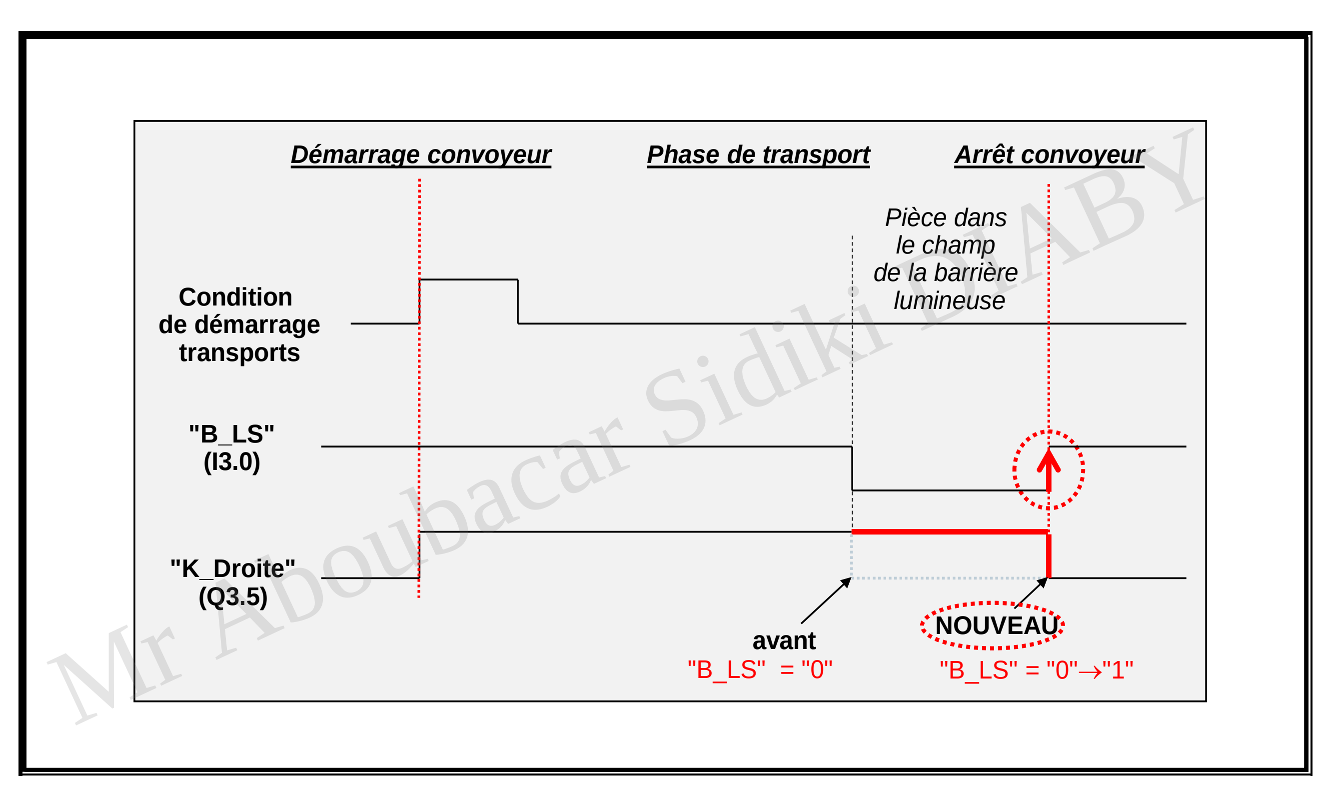
La réponse aux fronts du RLG permet de détecter si l'état d'une opération (ET dans l'exemple) est passé de '0' à '1' (front montant ou positif) ou de '1' à '0' (front descendant ou négatif). Si c'est le cas, l'instruction fournit le résultat RLG '1' pendant toute la durée d'un cycle. Ce résultat peut être directement relié à une autre opération ou être affecté en tant qu'état à un autre opérande (mémento « M_pulse_front_pos » (M16.1) dans l'exemple). Lors du cycle suivant, l'instruction fournit à nouveau le résultat '0, même si l'opération ET continue de fournir l'état '1'.

L'instruction compare le résultat courant ou le RLG de l'opération avec le RLG qu'ils avaient lors du cycle de programme précédent. Celui-ci est mémorisé à cet effet dans un mémento auxiliaire, dit « mémento de front » (« M_aux_gestion_front » dans l'exemple). Vous devez veiller à ce que l'état de ce mémento de front ne soit pas écrasé en un autre point du programme. Vous utiliserez pour ce faire un mémento de front distinct pour chaque réponse à un front, même si le même opérande (opération ET dans l'exemple) doit être réévalué plus tard, par ex. dans un autre bloc.

A propos des exemples si dessus:

- Avec les instructions P_TRIG et N_TRIG le résultat ou le RLG de l'évaluation du front est disponible à droite en sortie. Il peut être utilisé par la suite ou assigné à un opérande (dans l'exemple il est assigné aux opérandes « M_Pulse_front_pos » et « M_Pulse_front_neg ».

8.4.3. Exercice 3 : Intégrer la réponse aux fronts dans « FC_Convoyeur »



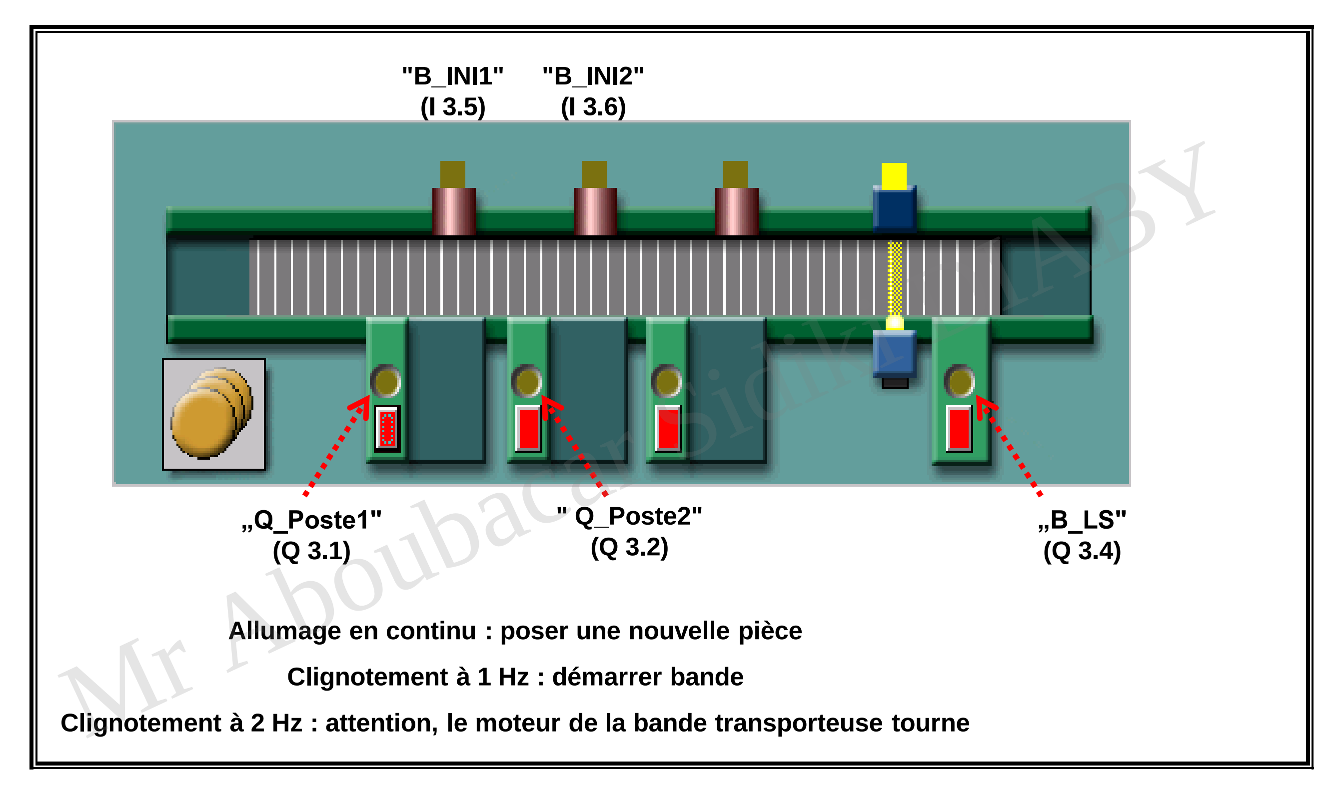
Enoncé

Lorsque « Q_en_service » (Q0.1 = « 1 ») est activé, les pièces du poste 1 ou 2 sont transportées jusqu'au FRANCHISSEMENT de la barrière lumineuse.

Marche à suivre

1. Programmez les modifications nécessaires dans « FC_Convoyeur » en reliant non plus le signal de la barrière lumineuse « B_LS » (I3.0) comme condition de remise à zéro de M16.4 (mémento pour moteur DROITE), mais le résultat de l'évaluation du front. Pour l'évaluation du signal de la barrière lumineuse, utilisez le mémento M16.0 comme mémento auxiliaire.
2. Chargez le bloc « FC_Convoyeur » modifié dans la CPU et vérifiez le fonctionnement correct du programme.

8.5. Enoncé : commander les témoins lumineux, mettre en service « FC_Signalisation »



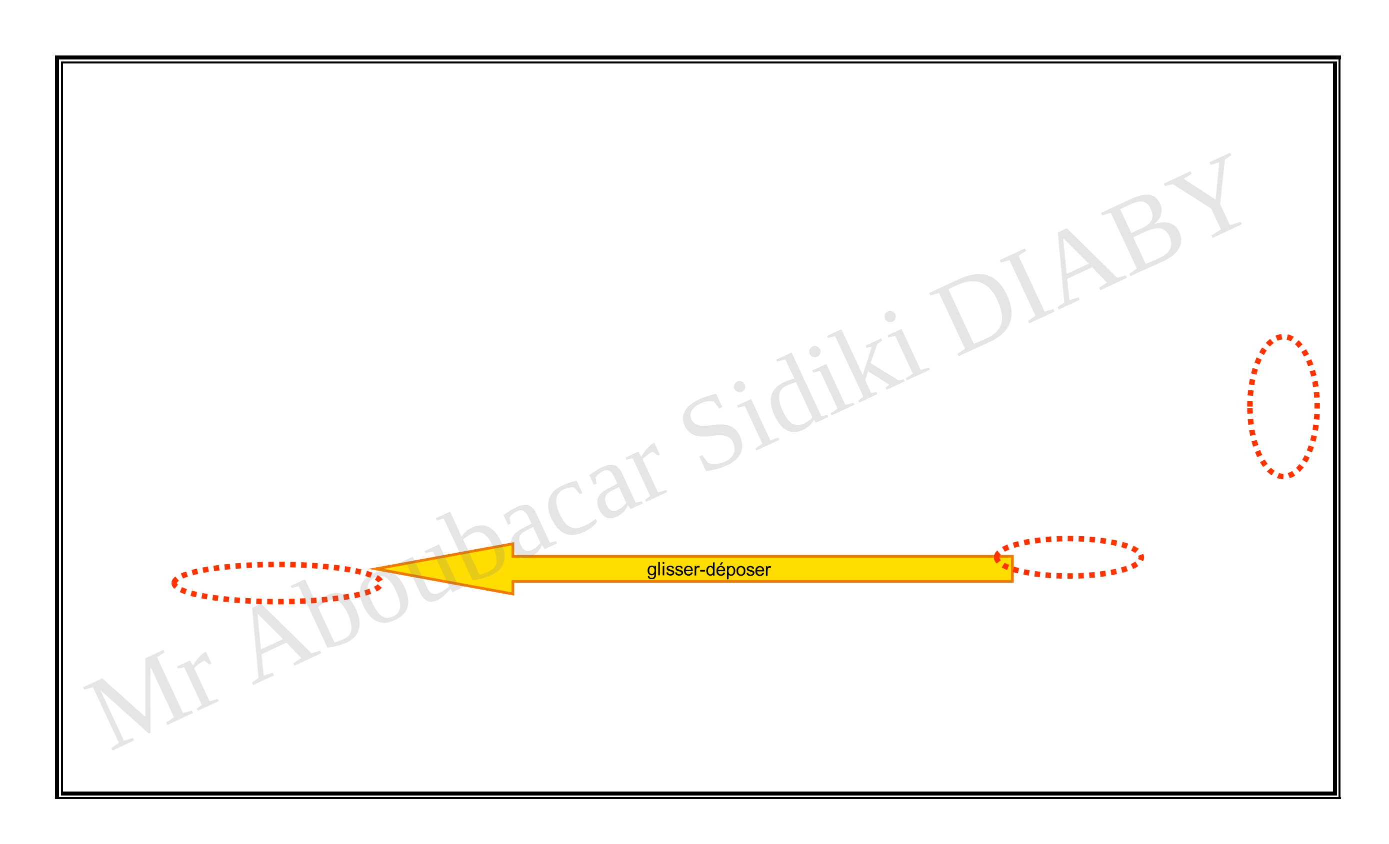
Enoncé

Lorsque « Q En Service » (Q0.1) est activé et tant qu'aucun « Q Defaut » (Q0.7) n'est présent, les témoins lumineux « Q_Poste1 » (Q3.1) et « Q_Poste2 » (Q3.2) doivent être commandés comme suit :

- Ils doivent être allumés en continu lorsque
  - une nouvelle pièce peut être déposée sur la bande (le moteur de la bande est à l'arrêt et les deux détecteurs de proximité sont libres)
  - clignotement à 1 Hz au niveau du poste où une pièce a été détectée par le détecteur de proximité correspondant, mais tant que la bande n'a pas encore été démarrée (si des pièces sont déposées sur la bande devant les deux détecteurs de proximité, aucun témoin lumineux ne doit être allumé)
  - clignotement à 2 Hz tant que le moteur tourne
- Le témoin lumineux au poste barrière lumineuse « B_LS » (Q3.4) clignote à 2 Hz tant que le moteur tourne.

Les fonctions décrites sont en partie déjà programmées dans le bloc « FC_Signalisation » , qui se trouve dans la bibliothèque globale « TIA_Pro1_Bib_fr ». Vous mettrez en service et compléterez ce bloc dans le prochain exercice.

8.5.1. Exercice 4 : Commander les LED de signalisation, programmation du bloc « FC_Signalisation »



Enoncé

Le bloc « FC_Signalisation » doit être copié dans le projet à partir de la bibliothèque globale « Bib_TIA_Pro1 », mis en service et complété.

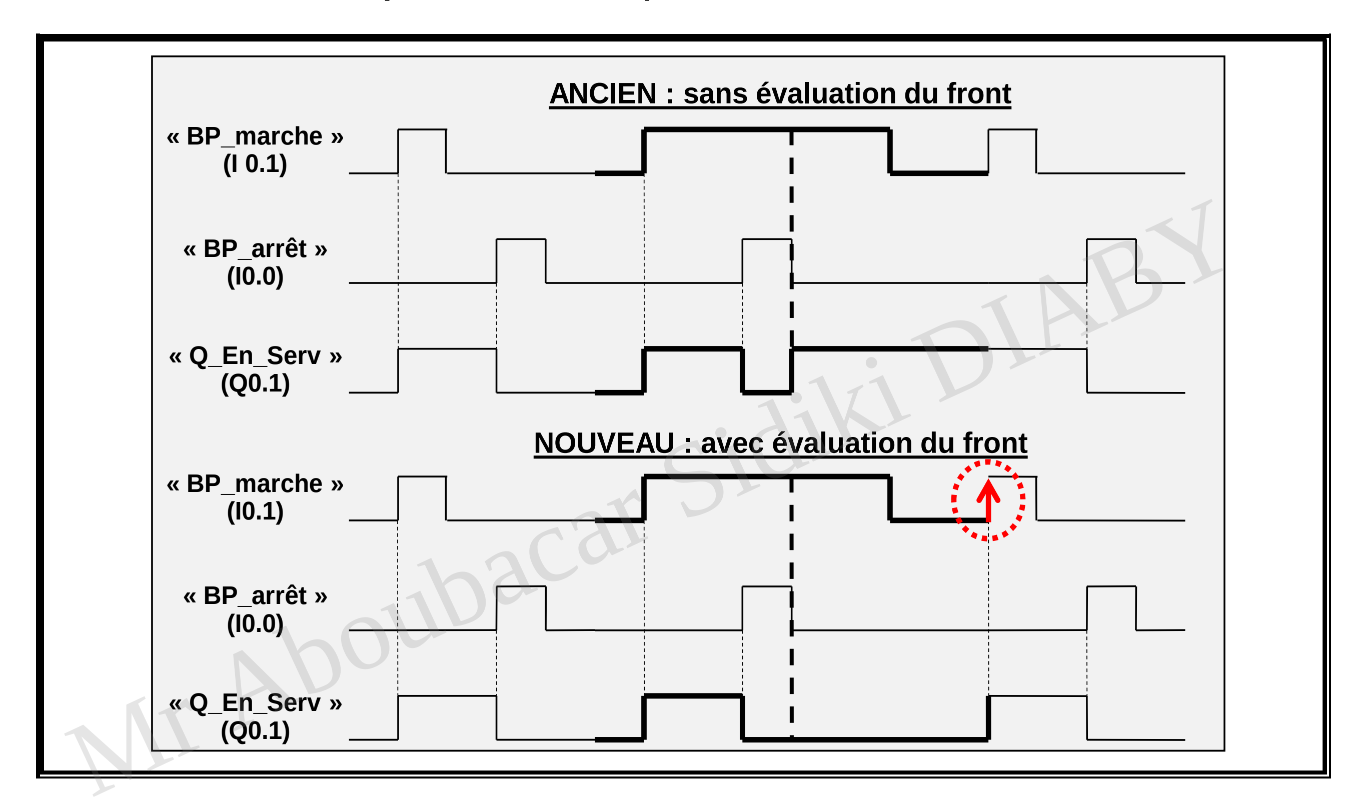
Marche à suivre

1. Copiez par glisser-déposer, comme indiqué dans la figure ci-dessus, le bloc « FC_Signalisation » de la bibliothèque globale « Bib_TIA_Pro1 » dans le dossier « Blocs de programme ».
2. Complétez le bloc de manière à ce que les témoins lumineux « Q_Poste1 » (Q4.1) et « Q_Poste2 » (Q4.2) ne soient pas commandés lorsque « Q Defaut » (Q0.7) est présent.
3. Programmez l'appel du « FC_Signalisation » dans l'« OB_Cyclique » .
4. Chargez tous les blocs modifiés dans la CPU et vérifiez le fonctionnement du programme.
5. Enregistrez votre projet.

Note:

Le bloc de programme complété peut être trouvé dans la bibliothèque dans le dossier « Chapitre 08 > prêt ».

8.5.2. Exercice complémentaire 6 : Optimiser le bloc « FC_En_Service » *



Fonction précédente de « FC_En_Service »

« Q_en_service » (Q4.1) est activé via le bouton du simulateur « BP_marche » (I0.1) et désactivé via le bouton du simulateur « BP_arret » (I0.0). En cas d'actionnement simultané des deux boutons, l'installation est ou reste désactivée. Cependant, si le bouton « BP_arret » est relâché alors que le bouton « BP_marche » reste actionné, l'installation se remet en route sans que l'on ait besoin d'actionner une nouvelle fois le bouton « BP_marche » (voir figure, diagramme fonctionnel supérieur « ANCIEN : sans évaluation de front »).

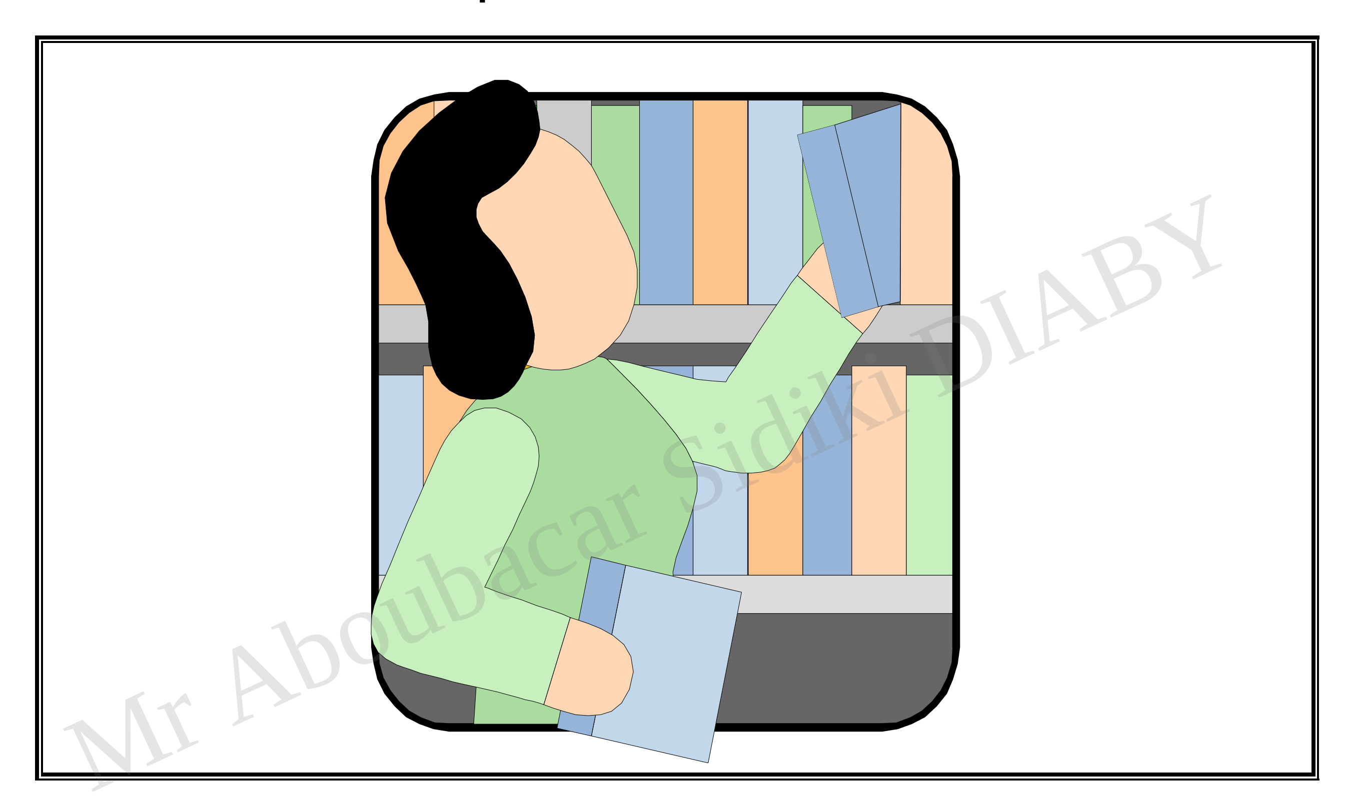
Enoncé

Etendez les fonctions de « FC_En_Service » avec une évaluation de front de manière à ce que toute activation de l'installation nécessite l'appui sur le bouton « marche » (voir figure, diagramme fonctionnel inférieur « NOUVEAU : avec évaluation du front »). Le critère d'activation de l'installation ne doit plus être l'appui sur le bouton « BP_marche » ni son signal « 1 », mais l'opération d'activation ou le « front montant » du signal du bouton « BP_marche ».

Marche à suivre

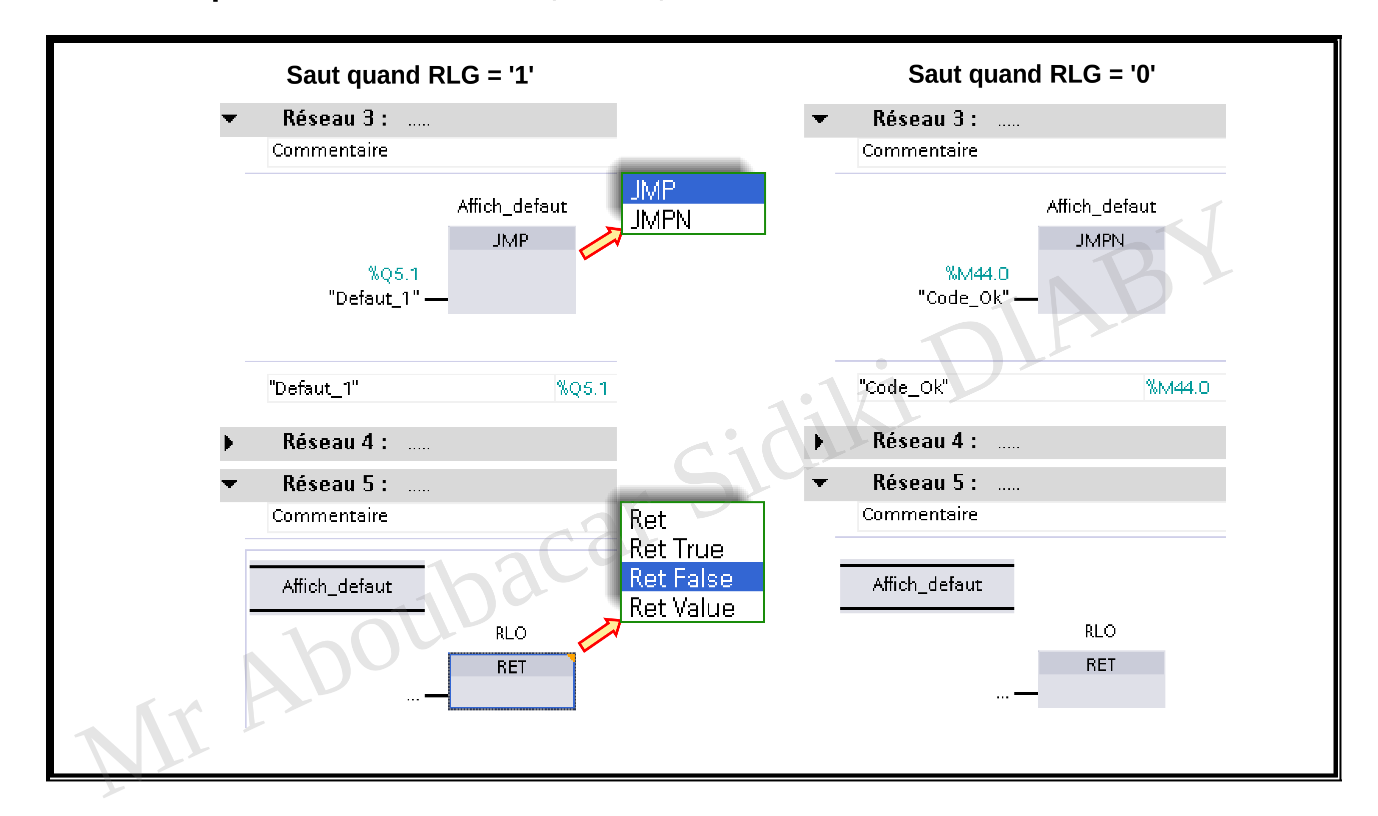
1. Ajoutez aux conditions de mise à 1 pour « Q_en_service » (Q0.1) une évaluation du front du bouton « BP_marche » (I0.1). Utilisez pour l'évaluation du front le mémento « M_front_BP_Marche» (M15.1) comme mémento auxiliaire.
2. Chargez le bloc « FC_En_Service » modifié et vérifiez s'il remplit les fonctions voulues.

8.6. Informations complémentaires





8.6.1. Opérations de saut JMP, JMPN, RET



Opérations de saut JMP et JMPN

Les opérations de saut JMP et JMPN permettent d'interrompre le traitement linéaire du programme au sein d'un bloc et de le poursuivre dans un autre réseau. Un repère de saut (étiquette), qui identifie aussi le réseau de destination du saut, est indiqué au-dessus de l'opération de saut. Le repère de saut et la destination du saut doivent se trouver dans le même bloc. La destination du saut ne doit être présente qu'une seule fois dans ce bloc. Chaque repère de saut peut faire l'objet de sauts à partir de plusieurs points. Les sauts peuvent se faire vers l'avant ou vers l'arrière, c'est-à-dire vers des réseaux de numéro supérieur ou inférieur.

- **JMP :**
  Si le RLG est égal à « 1 », le saut vers le réseau de destination est exécuté ; si le RLG est égal à « 0 », le saut vers le réseau de destination n'est pas exécuté et le traitement linéaire du programme se poursuit.
- **JMPN :**
  Si le RLG est égal à « 0 », le saut vers le réseau de destination est exécuté ; si le RLG est égal à « 1 », le saut vers le réseau de destination n'est pas exécuté et le traitement linéaire du programme se poursuit.

Opération de retour RET

L'opération RET met fin au traitement du bloc. SI l'entrée de l'instruction est assignée alors c'est une « fin de bloc conditionnelle » et se poursuit uniquement par un avortement du bloc quand le scan est complété.

Chaque bloc a une sortie de validation "ENO" à travers laquelle les informations concernant l'exécution du bloc sans erreur peuvent être communiquées au bloc appelant. Avec l'aide de la fonction RET, la sortie de validation peut être écrite.

- **Ret** (RLO = le résultat d'une opération logique. Le bloc appelant fournit le statut du signal "1" depuis l'instruction RET, comme instruction conditionnelle, qui est seulement exécuter lorsque la condition est VRAI.

- **Ret True** ou **Ret False** (Les valeurs respectives des constantes, VRAI ou FAUX, est fournit au bloc appelant.)
- **Ret Value** (La valeur de la variable Booléenne <Operand>, qui est spécifié au dessus de l’instruction, est fournit au bloc appelant.)



Sommaire 9

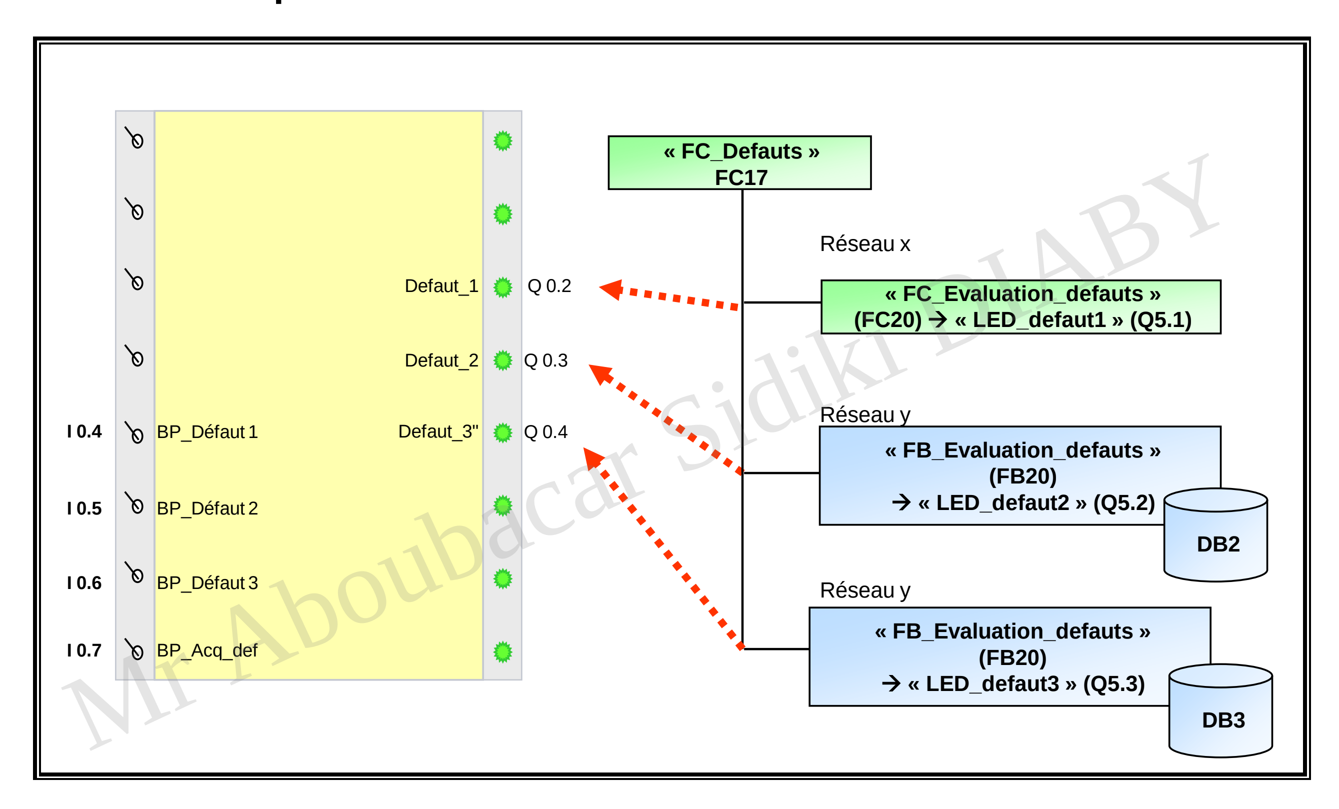
9. Fonctions et bloc de fonction

A l'issue de ce chapitre, vous allez...

- … **connaître l'utilité des variables temporaires**
- ... **connaître l'utilité des blocs paramétrables**
- ... **savoir programmer des fonctions paramétrables et leur appel**
- ... **connaître la différence entre les fonctions (FC) et les blocs fonctionnels (FB)**
- ... **connaître l'utilité des variables statiques**
- ... **savoir déclarer des variables statiques et les utiliser dans un programme**
- ... **savoir programmer des blocs fonctionnels paramétrables et savoir les appeler dans le programme**



9.1. Énoncé : Evaluation de la signalisation des défauts avec des blocs paramétrables



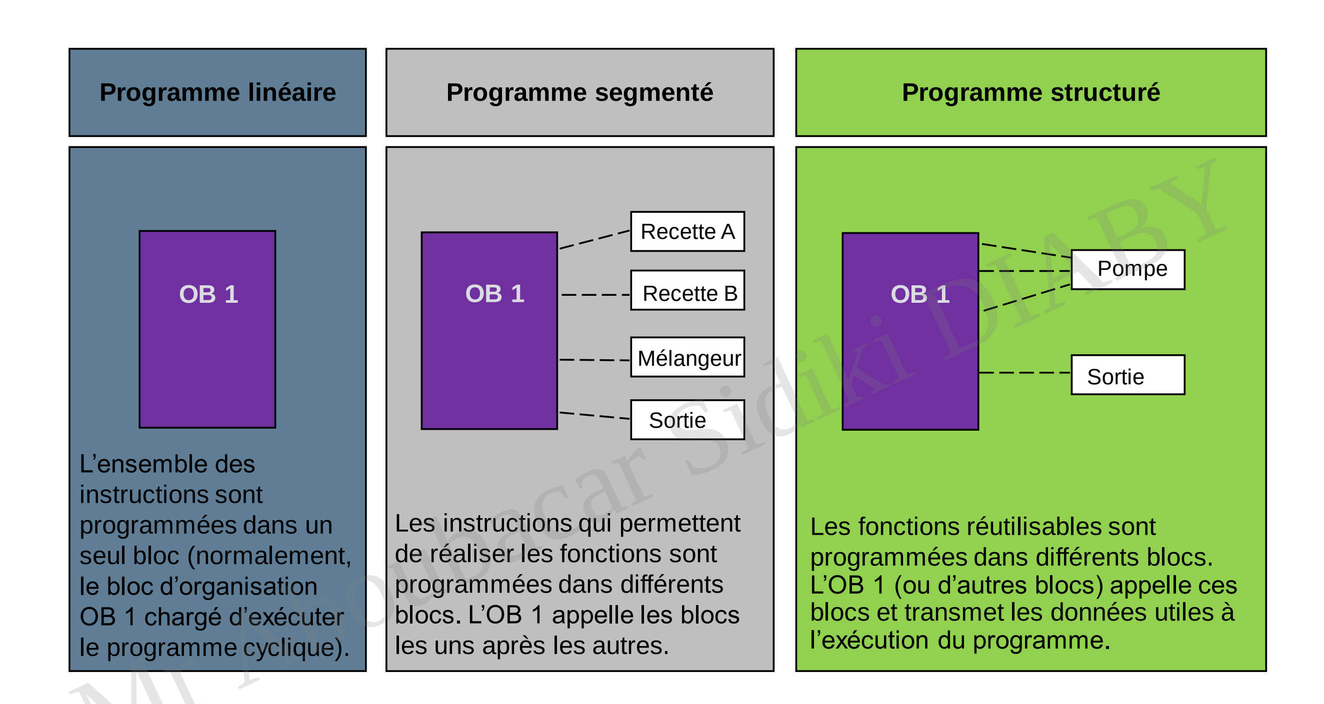
Enoncé

Indépendamment des fonctions actuelles de la maquette de bande transporteuse, 3 signaux de défaut différents doivent désormais être évalués comme suit :

Lorsqu'un signal de défaut est émis sur les entrées I0.4 à I0.6, les LED Q0.2 à Q0.4 commencent à clignoter.

L'entrée I0.7 (« BP_Acquittement_def ») permet d'acquitter l'ensemble des signaux de défaut (acquittement groupé des défauts). Si la signalisation du défaut perdure après acquittement du défaut, la LED reste allumée en continu. Si la signalisation du défaut disparaît, la LED s'éteint.

9.2. Programmation structurée



Programme linéaire

Le programme entier est compris dans un bloc de programme continu (Cycle programme OB) qui est automatiquement appelé par le système. Ce model ressemble au relai de contrôle qui ont été remplacé par les systèmes automatisés (Contrôleur Logique Programmable). La CPU traite les instructions individuelles les unes après les autres.

Programme structuré avec des blocs

Le programme est divisé entre plusieurs blocs, qui chacun contiennent le programme pour résoudre une tâche partiel. Une structuration supplémentaire via des réseaux est possible dans un bloc. Vous pouvez générer des modèles de réseau pour des réseaux du même type. Normalement, un bloc Organisation appelé cycliquement contient des instructions qui appellent les autres blocs dans une séquence définie.

Programme Structuré

Un programme structuré contient des blocs paramétrables qui sont crées de façons à être utilisés de manières universelles. Quand un bloc paramétrable est appelé, des paramètres de courant sont transmis (par exemple, les adresses spécifiques des entrées et sorties ainsi que les valeurs des paramètres….).

Exemple:

- Un « bloc pompe » contient les instructions pour le contrôle de la pompe.
- Le bloc programme, qui est responsable du contrôle des pompes spéciales, appel le « bloc pompe » et lui fournit les informations de quel pompe doit être contrôler avec quel paramètre.

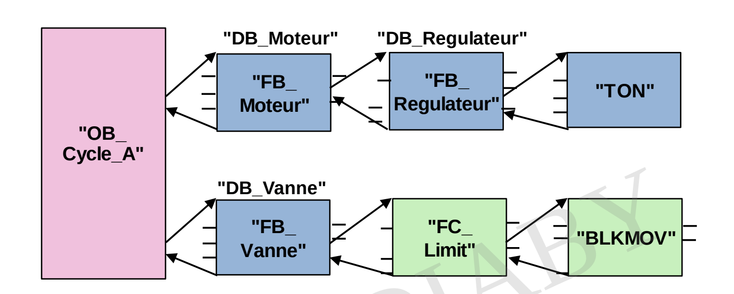
9.2.1. Bloc Modulaire et réutilisable

Modularisation de l'ensemble de la âche :

Les tâches partielles sont exécutées dans des blocs différents

Le paramétrage permet une utilisation flexible

- Exemple : cycle de forage avec profondeur paramétable

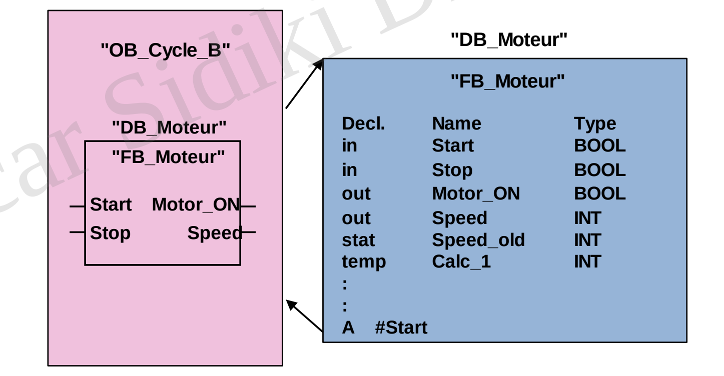


Utilisation multiple de blocs :

Les blocs peuvent être appelés un nombre de fois illimité

Restrictions :

- Pas d'accès aux opérandes globaux dans le bloc
- Communication uniquement par paramètres



Modularisation de l'intégralité de la tâche

L'abstraction est la base pour résoudre les problèmes complexes, dans lequel nous nous concentrons sur l'aspect fondamentaux d'un programme dans tous les niveaux d'abstractions et ignorons tous les détails qui ne sont pas essentiels. L'abstraction nous aide à diviser les tâches complexes en tâches partielles qui peuvent être alors résolus individuellement.

Bloc paramétrable (réutilisable)

STEP7 supporte le concept de la modularisation avec son modèle de bloc. Les tâches partielles qui résultent de la division d'une tâche complète sont assignées aux blocs dans lesquelles l'algorithme nécessaire et les données pour résoudre le problème partiel sont stockés. Les blocs STEP7 tels que les fonctions (FC) et les blocs fonctions (FB) peuvent être affectés de paramètres afin que les concepts de la programmation structurés puissent être implémentés avec eux. Cela signifie :

- Les blocs de résolution des tâches partielles implémentent leur propre gestion de donnée avec l'aide des variables locales.
- Les blocs qui communiqué avec « le monde extérieur », c'est-à-dire, avec les capteurs et les actionneurs du contrôle de processus ou avec d'autres blocs du programme utilisateurs, exclusivement à partir les paramètres des blocs. Aucun accès aux opérandes globaux tels que les entrées, sorties, bits mémoire ou les variables dans les DBs peut être fait à partir de la section d'instruction des blocs.

Mr Aboubacar Sidiki DIABY

Avantages

- Les blocs pour les tâches partielles peuvent être créés et testés indépendamment des autres.
- Les blocs peuvent être appelés aussi souvent que requis dans différents emplacements du programme avec des ensembles de paramètres différent. Ils peuvent donc être réutilisés.
- Les blocs « réutilisables » pour certaine tâches sont livrés dans des bibliothèques prédéfinies.

9.2.2. Opérandes locaux et globaux

Variables globales (valides dans tout le programme)	*Variables locales (valides dans un seul bloc)*
• MIE / MIS ***• Périphéries d'E/S*** ***• Mémentos*** ***• Variables d'un DB*** *(sera traité dans un autre chap.)* ***• Temporisation S5 *)*** ***• Compteur S5 *)*** ***• Constantes*** **) pas pour le S7-1200*	***Opérandes formels*** ***• Interface d'échange entre bloc appelant et appelé*** ***• mémorisation temporaire dans la pile L pour les FC et mémorisation dans le DB d'instance pour les FB*** ***• utilisation dans les FC / FB*** ***Variables temporaires*** ***• sont écrasées après le traitement*** ***• mémorisation temporaire dans la pile L*** ***• utilisation dans les OB / FC / FB*** ***Variables statiques*** ***• conservent leurs valeurs après le traitement du bloc*** ***• stockage permanent dans le DB d'instance*** ***• utilisation uniquement dans les FB*** ***Constantes*** ***• uniquement accessibles en lecture et en symbolique*** ***• pas d'occupation mémoire*** ***• utilisation dans les OB / FC / FB***

Variables globales

Les variables globales sont valides dans l'ensemble du programme S7. Ainsi ces variables sont accessibles pour tous les blocs de code.

Parmi les variables globales figurent les opérandes classiques pour API tels que les entrées, les sorties, les mémentos, les temporisations SIMATIC, les compteurs SIMATIC, les constantes et les variables déclarées dans des blocs de données globaux.

Variables locales

Les variables locales ne sont valides que dans le bloc dans lequel elles ont été saisies dans la section de déclaration. Ainsi ces variables ne sont accessibles que pour ce bloc.

- Variables temporaires

  Les variables temporaires peuvent être déclarées dans tous les blocs de code (OB, FC, FB), elles sont gérées dans la pile des données locales de la CPU. Ainsi elles ne conservent leurs valeurs que pour la durée de traitement du bloc. C'est pour cette raison qu'il faut, pour le cycle en cours, un accès en écriture avant l'accès en lecture pour une variable temporaire. Elles ne conviennent pas par exemple pour servir de variables auxiliaires dans la gestion des fronts ou pour la mémorisation d'un comptage de pièces. Elles sont principalement utilisées pour la sauvegarde de résultats intermédiaires comme par ex. lors de calculs complexes ou de conversions de format.

- Variables statiques

  Les variables statiques ne peuvent être déclarées que dans les FB et sont mémorisées dans le bloc de données d'instance associé. Ainsi ces variables conservent leurs valeurs même après le temps de traitement du FB.

- Opérandes formels

  Ils forment l'interface entre le bloc appelant et le bloc appelé. Ils sont utilisés pour réaliser l'échange des données entre le bloc appelant et le bloc appelé.

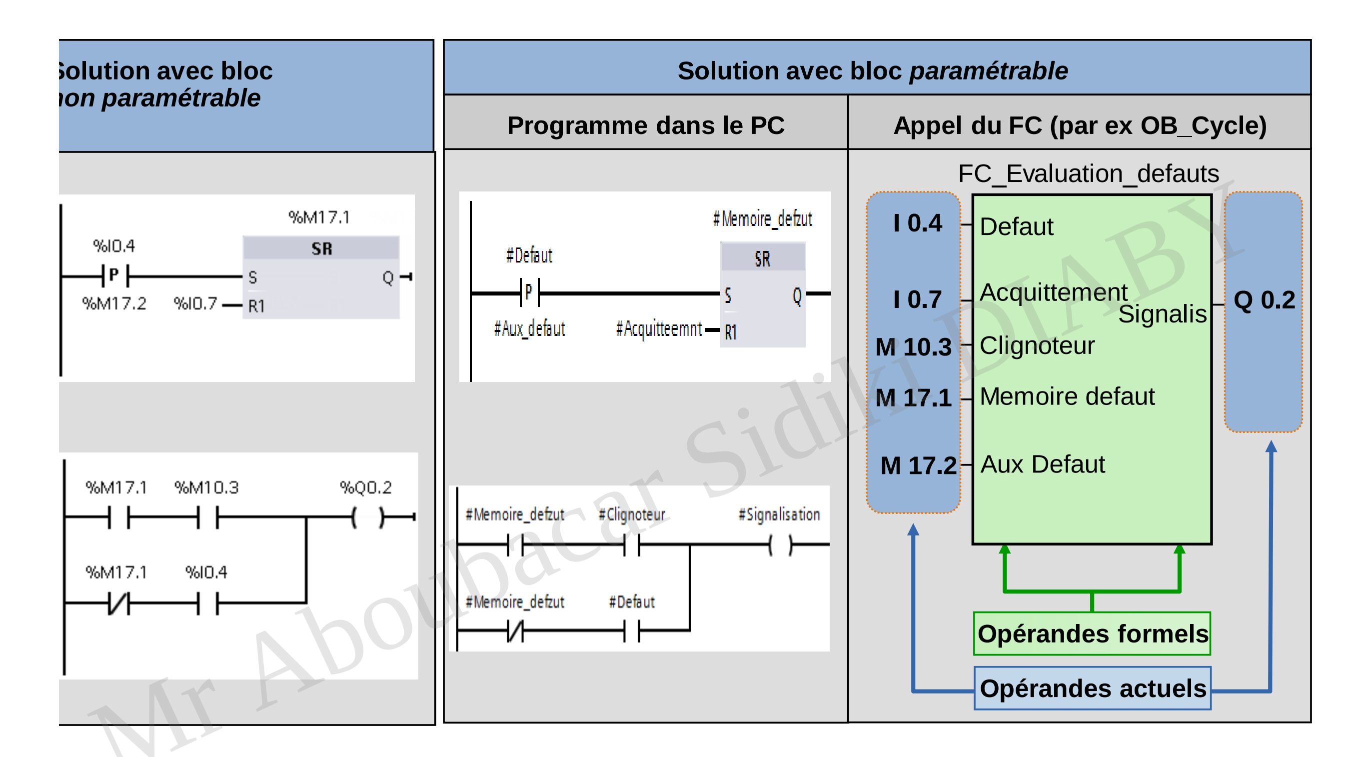
- Constantes

  Ce sont des valeurs figées seulement accessibles en lecture qui n'ont pas d'affectation mémoire.

- Constants

  Constants are fixed values which have a read-only access and which do not take up any memory space.

9.3. Solution avec un bloc paramétrable



Application

Pour les séquences de programme récurrentes, on peut utiliser des blocs paramétrables qui offrent les avantages suivants :

- Le programme n’est défini qu’une seule fois, ce qui réduit considérablement le temps de programmation.
- Le bloc est stocké une fois pour toutes dans la mémoire utilisateur, libérant ainsi un espace mémoire important.
- Le bloc, ou la fonction réalisée avec ce bloc, peut être appelé aussi souvent que nécessaire, chaque fois avec d’autres opérandes. Pour ce faire, des opérandes effectifs sont affectés aux opérandes formels (paramètres d’entrée, de sortie ou d’E/S) à chaque appel.

Traitement du programme

Lors du traitement du bloc, l’opérande formel est remplacé pour l’opération « *A #defaut* » par l’opérande actuel affecté lors de l’appel.

Si l’entrée I1.1 a été attribuée comme opérande actuel pour le paramètre *defaut* lors de l’appel du bloc, l’opération « A I 1.1 » est finalement exécutée au lieu de l’opération « A *#defaut* ».

Blocs paramétrables

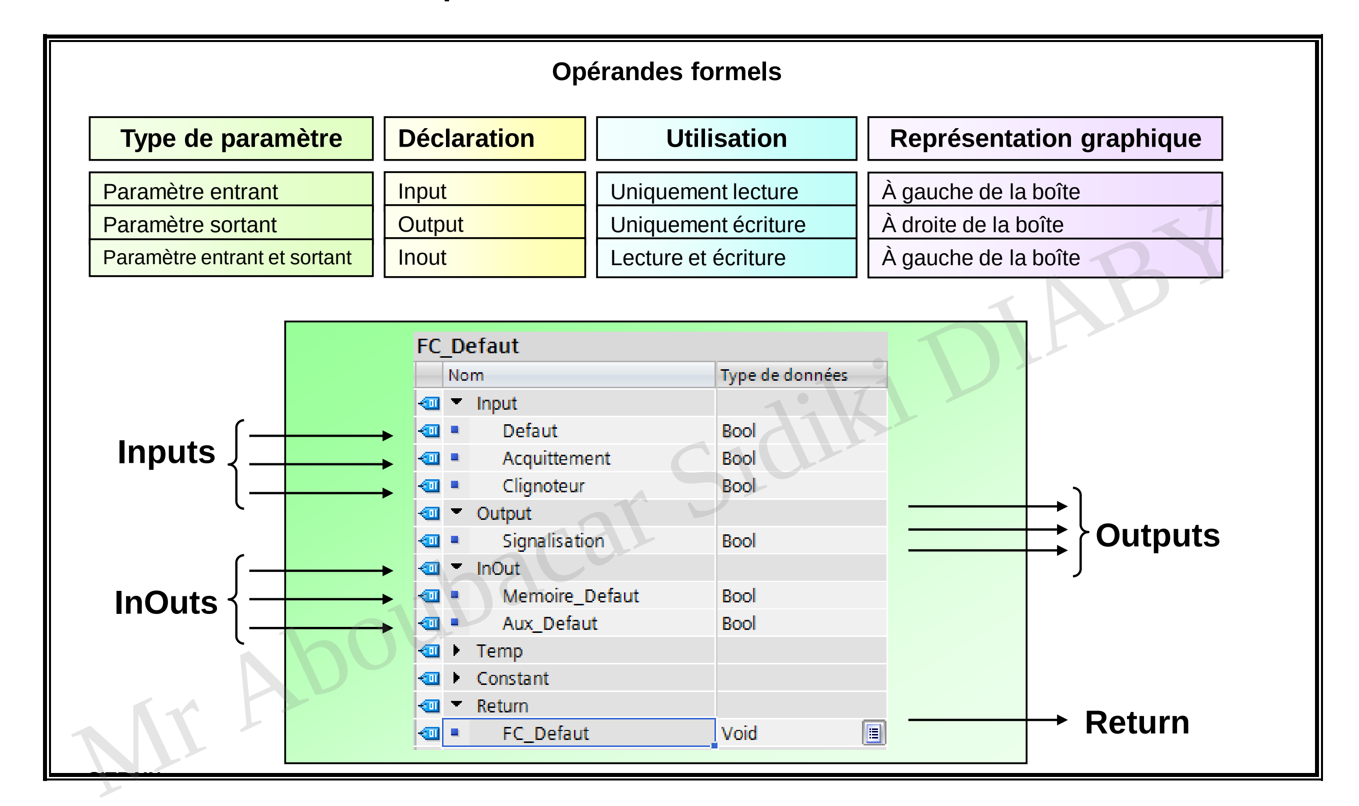
Les blocs paramétrables peuvent être du type FC ou FB. Les blocs d’organisation ne sont pas paramétrables parce qu’ils sont appelés par le système d’exploitation, ce qui interdit la transmission d’opérandes actuels au moment de l’appel du bloc.

Exemple

Même si la fonction Analyse de défaut ou Signalisation de défaut est utilisée plusieurs fois dans l’installation, le bloc FC 20 n’est programmé qu’une seule fois.

Pour réaliser 2 analyses de défaut différentes, il suffit d’appeler deux fois le bloc FC20 et de lui affecter chaque fois des opérandes actuels différents.

9.3.1. Déclaration des opérandes formels



Type de paramètre	Déclaration	Utilisation	Représentation graphique
Paramètre entrant	Input	Uniquement lecture	À gauche de la boîte
Paramètre sortant	Output	Uniquement écriture	À droite de la boîte
Paramètre entrant et sortant	Inout	Lecture et écriture	À gauche de la boîte

Opérandes formels

Avant de pouvoir établir le programme pour le bloc paramétrable, il faut définir les opérandes formels dans la partie déclarative.

Type de paramètres

La table de cette diapositive montre les trois types de paramètre possibles et leur utilisation. Il ne faut pas oublier de déclarer comme paramètres InOut, les opérandes nécessitant un accès en lecture et en écriture.

Interface

Les paramètres Input, Output et InOut constituent l'interface d'un bloc. Le paramètre RETURN est un paramètre Output supplémentaire, défini selon la norme CEI 61131-3 avec un nom fixe et contenu dans l'interface seulement dans le cas des FC.

Même si elles figurent sous « Interface », les variables TEMP ne font pas partie de l'interface du bloc car elles n'ont pas à apparaître lors de l'appel d'un bloc.

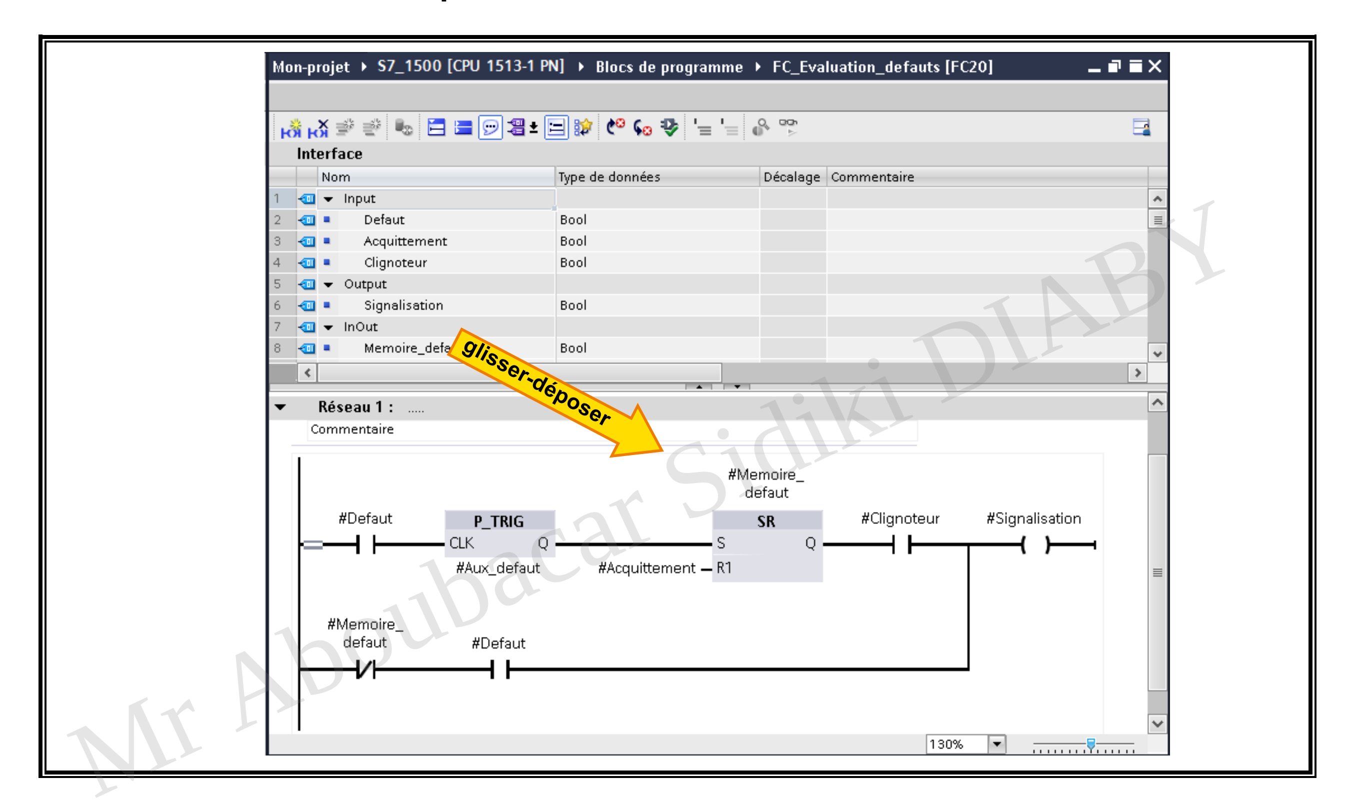
Exemple de FC20

La partie inférieure de cette diapositive montre la partie déclarative ou l'interface du bloc FC 20 « FC_evaluation_defaut » (voir page précédente). Comme l'opération FP accède aux paramètres #Memoire_defaut et #aux_defaut en lecture et en écriture, ces derniers doivent être déclarés en tant que paramètres InOut.



Attention !
Les paramètres définis dans un bloc (Input, Output et InOut, et non Temp) constituent l'interface de ce bloc vers « l'extérieur », c'est-à-dire qu'ils sont « visibles » ou significatifs dans d'autres blocs appelant ce bloc. Si l'interface d'un bloc est modifiée par l'effacement ou l'ajout ultérieur de paramètres, les appels doivent être actualisés ou corrigés en conséquence dans tous les blocs dans lesquels l'appel de ce bloc est programmé.

9.3.2. Edition d'un bloc paramétrable



Nota

L'opérande formel peut être écrit indifféremment en majuscules ou en minuscules. Le caractère « # » est inséré automatiquement devant le nom par l'ordinateur. Ce caractère signale qu'il s'agit d'une variable locale définie dans la partie déclarative de ce bloc.

En programmation CONT / LOG, il se peut que le nom occupe plus d'une ligne. Le réglage s'effectue dans l'éditeur de programme :

Outils → Paramètres →Programmation API→ Onglet "CONT/LOG" → Largeur du champ d'opérandes

Mnémonique

1. Si l'opérateur utilise un nom symbolique lors de l'édition d'un bloc, l'éditeur explore d'abord la table de déclaration des variables. Si le mnémonique existe, il est intégré dans le programme précédé d'un # en tant que variable locale.
2. Si le mnémonique est introuvable à cet emplacement, l'éditeur recherche des mnémoniques globaux dans la table des mnémoniques. Si le mnémonique existe, il est mis entre guillemets puis inséré dans le programme.
3. Si le même nom symbolique a été attribué comme mnémonique global (dans la table des mnémoniques) et comme mnémonique local (dans la partie déclarative des variables), l'éditeur intègre toujours la variable locale.

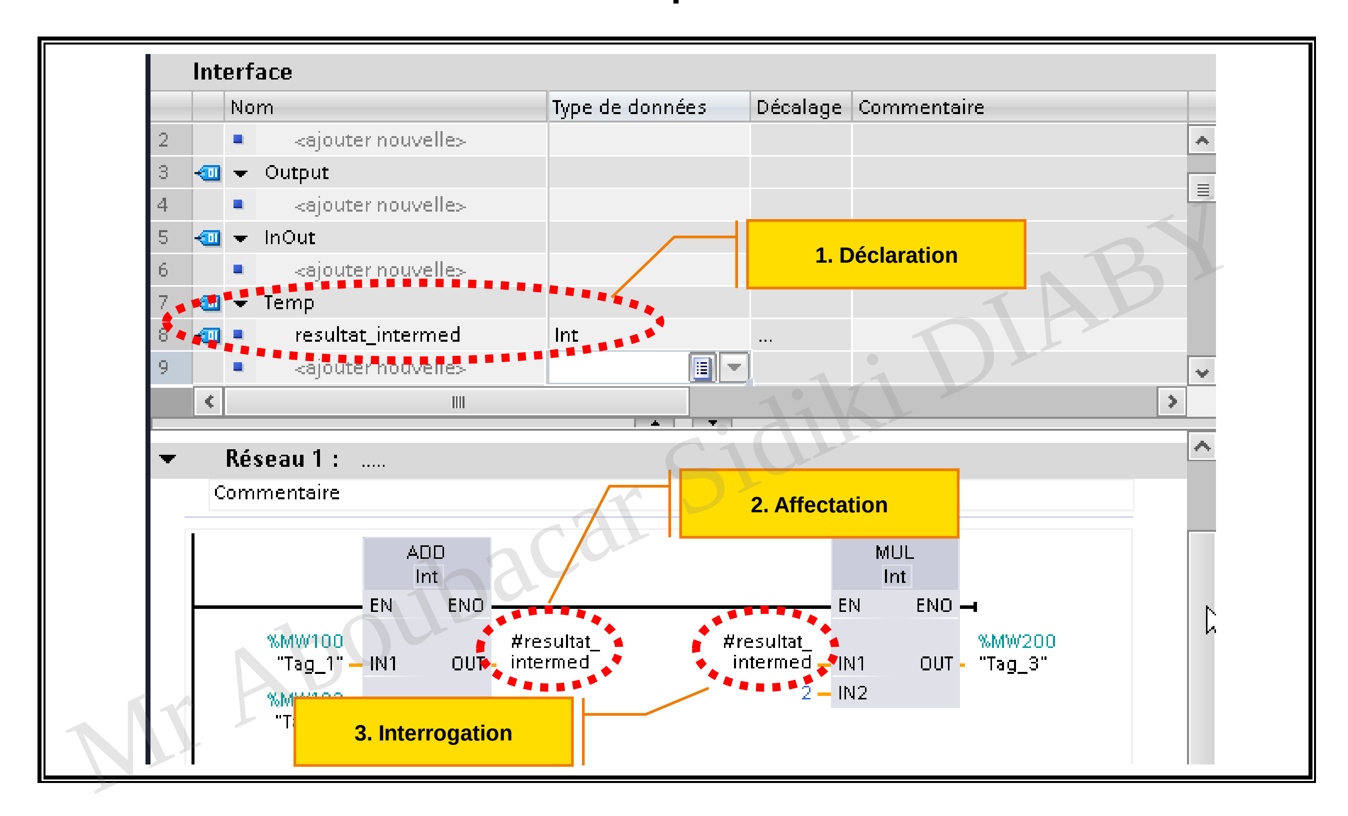
   Cependant, si l'opérateur veut utiliser le mnémonique global, il doit mettre le mnémonique entre guillemets au moment de la saisie.

Glisser&déposer

Tout comme les variables globales, par ex. les variables API, les variables locales peuvent être déplacées par glisser&déposer depuis l'interface du bloc vers la partie déclarative de l'éditeur et être positionnées à l'endroit souhaité.

9.4. Utilisation des variables temporaires





Déclaration

Les variables sont définies dans la partie déclarative du bloc. Pour les déclarer, indiquez un nom de variable et le type de données correspondant.

A ce titre, il n'est pas possible d'affecter une valeur initiale à la variable. Après enregistrement du bloc, l'adresse relative de la variable dans la pile L s'affiche dans la colonne « Adresse ».

Accès

Au début du traitement du bloc, toutes les variables temporaires ont une valeur <u>indéterminée</u>. Lors du traitement d'un bloc avec des variables temporaires, il convient donc de veiller à ce qu'une valeur définie soit tout d'abord affectée aux variables avant de les interroger.

Dans notre exemple, le résultat de l'addition est affecté aux variables temporaires *#resultat_intermed* avant l'interrogation pour la multiplication.

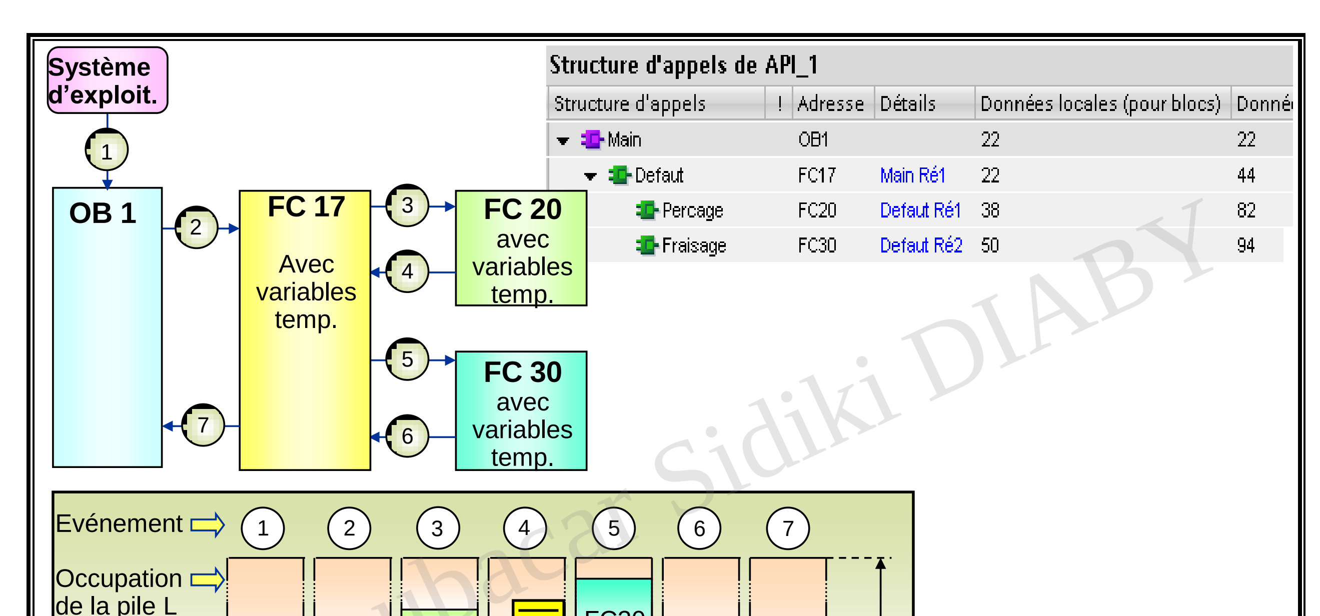
Bien que l'accès absolu aux variables temporaires soit également possible (T LW0), il n'est pas recommandé car il nuit à la lisibilité du programme.

Remarque

Si les noms d'opérandes sont précédés du caractère spécial #, il s'agit d'opérandes locaux (paramètres ou variables locales), qui doivent être déclarés dans la partie déclarative du bloc, valides uniquement au sein du bloc dans la partie déclarative duquel ils ont été déclarés.

Le caractère # est automatiquement inséré par l'éditeur de programme.

9.4.1. Espace occupé dans la pile de données locales



Espace occupé dans la pile L (pile de données locales)

Pour chaque niveau de traitement du programme ou pour chaque classe de priorité (comme par exemple l'OB 1 avec tous les blocs appelés), une pile de données locales distincte (pile L) est réservée (un segment de taille définie est alloué sur la pile L de la CPU).

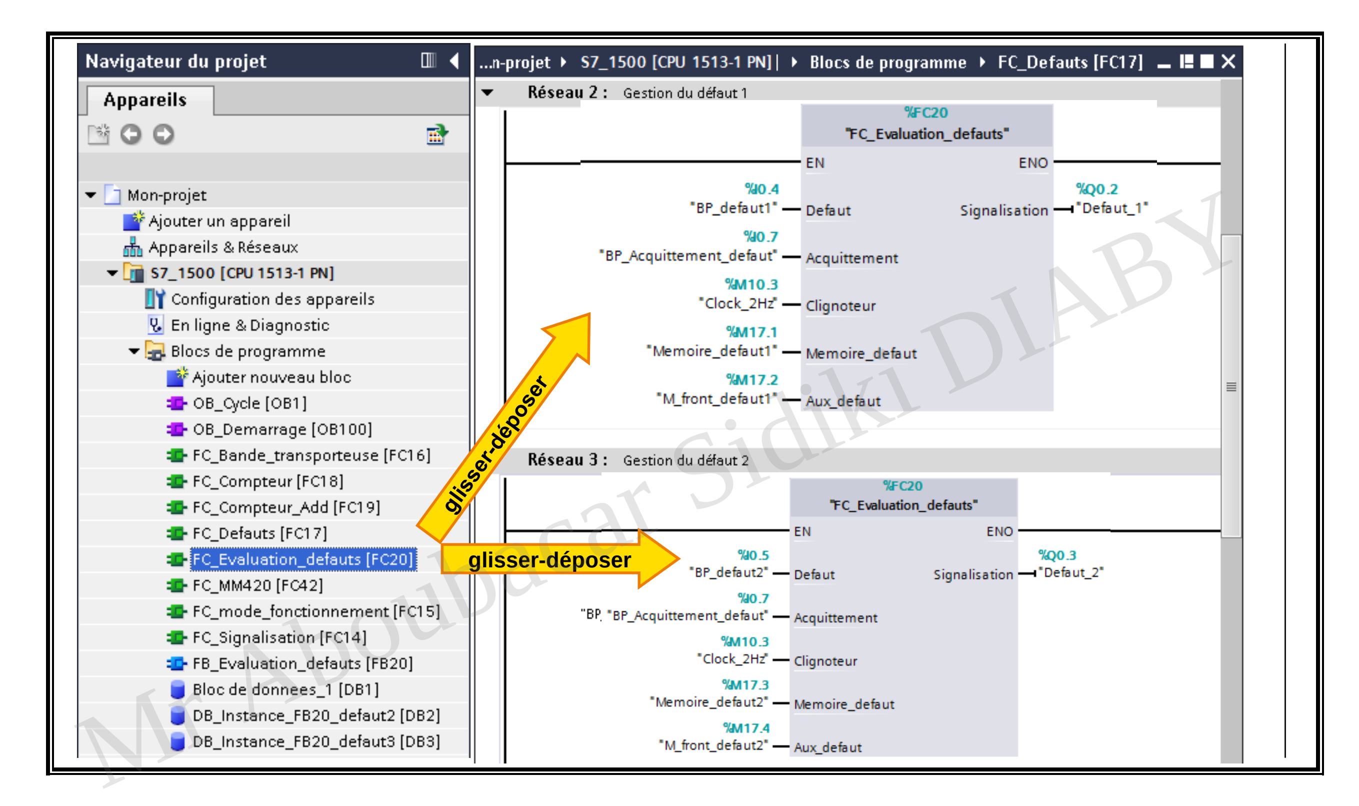
Cette pile L recevra aussi bien les variables locales de l'OB 1 lui-même que les variables locales temporaires des blocs (FC et FB) appelés à partir de l'OB1.

L'espace occupé dans la pile L par un programme S7 peut être visualisé à l'aide de la fonction « *Données de références* » qui sera traitée plus en détail au chapitre « Recherche d'erreurs ».

Nota

Si le nombre maximal de données locales est atteint (débordement de la pile L) lors de l'exécution du programme dans la CPU, la CPU se met à l'arrêt. La cause « Erreur d'allocation des données locales » est inscrite dans le tampon de diagnostic.

9.5. Appel d’un bloc paramétrable



Appel d’un bloc

Pour programmer l'appel d'un bloc paramétrable, il faut rechercher le bloc souhaité dans le dossier « Blocs de programme » et le déplacer par glisser-déposer dans la partie Instructions du bloc appelant.

Nota

Lors de l'appel d'une fonction paramétrable (FC), un opérande actuel doit obligatoirement être affecté à chaque opérande formel.

Exception :

Dans les langages de programmation graphiques CONT et LOG, les paramètres EN et ENO insérés automatiquement par l'éditeur peuvent être laissés libres

Paramétrage

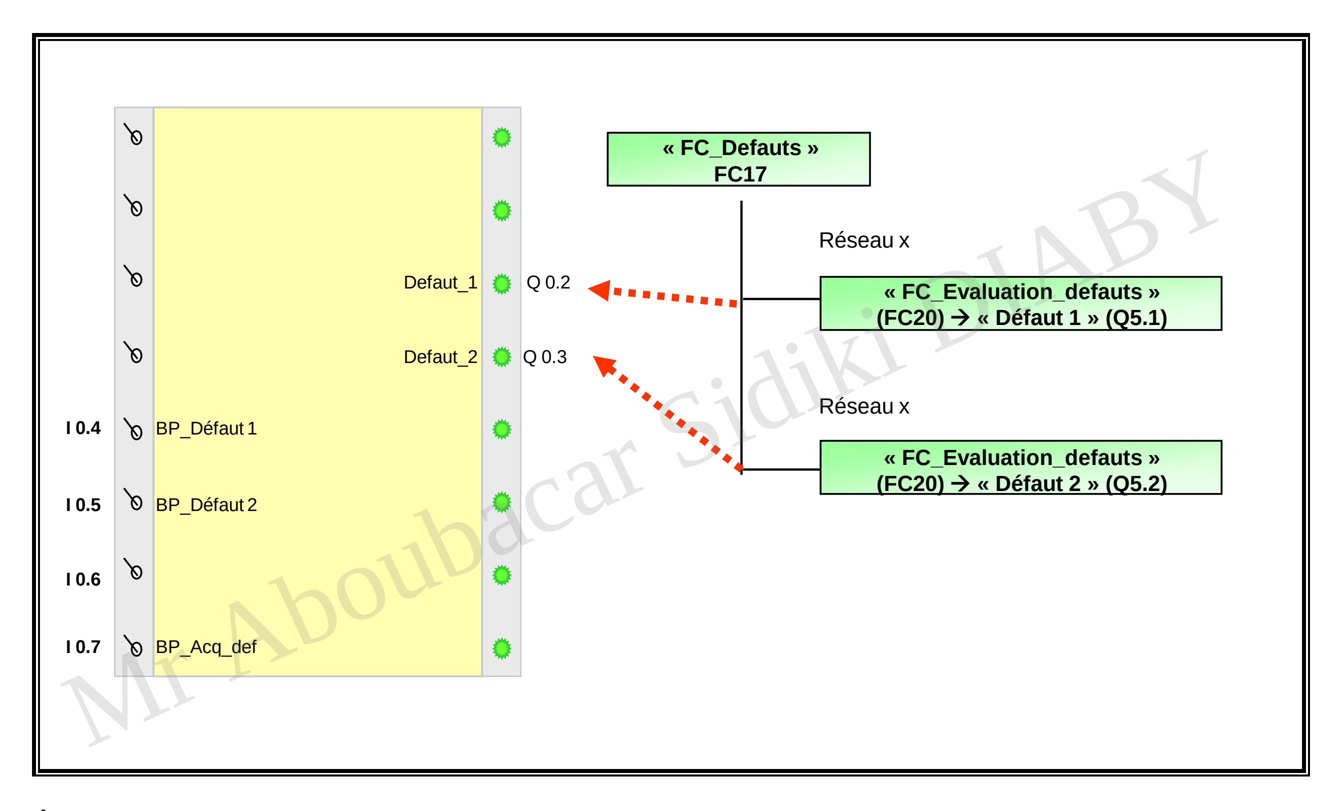
Les opérandes actuels autorisés sont tous les opérandes globaux et locaux dont le type de données correspond à l’opérande formel du bloc appelé.

Les opérandes actuels peuvent être transmis avec une adresse absolue ou avec des noms symboliques, tels qu’ils sont définis dans la table des mnémoniques ou dans la partie déclarative du bloc appelant.

Transfert de paramètres

Il est également possible d'effectuer un « transfert de paramètres », consistant à transmettre au bloc appelé les opérandes formels du bloc appelant comme opérandes actuels. Si les paramètres sont de type complexe, le transfert est également autorisé, mais avec des restrictions qui seront abordées dans un cours avancé.

9.6. Énoncé : Programmer et appeler la fonction paramétrable « FC_Evaluation_defauts »



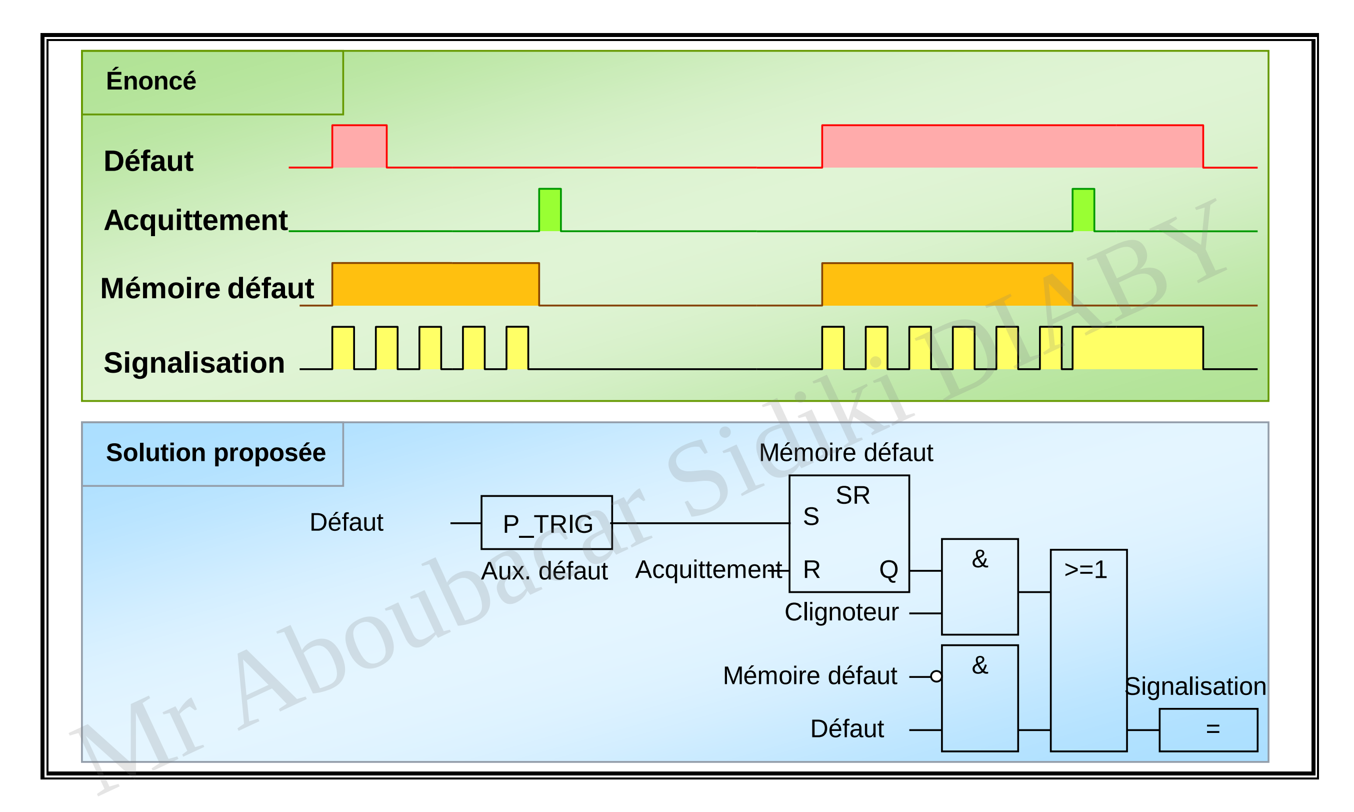
Énoncé

Lorsqu'un signal de défaut est émis aux entrées I0.4 ou I0.5, la LED Q0.2 ou Q0.3 correspondante se met à clignoter.

L'entrée I0.7 (« BP_Acquittemnt_def ») est un bouton d'acquittement groupé pour l'ensemble des signaux de défaut. Si le signal de défaut perdure après acquittement, la LED s'allume alors en continu. Si le signal de défaut n'est plus émis, la LED s'éteint.

La fonction requise doit être programmée dans le bloc paramétrable « FC_Evaluation_defauts » (FC20) qui doit être appelé deux fois dans le bloc « FC_Defauts » (FC17) pour évaluer les deux signaux de défaut.

9.6.1. Exemple de signalisation d'un défaut



Enoncé

La présence de défauts doit être signalée par un voyant lumineux sur le pupitre de commande. Lorsque le signal passe de 0 → 1 à l'entrée « defaut », le voyant de signalisation doit clignoter à 2 Hz.

Dès que le défaut est éliminé, le voyant s'éteint ; si malgré l'acquittement, le défaut perdure, l'éclairage du voyant reste fixe jusqu'à l'élimination du défaut.

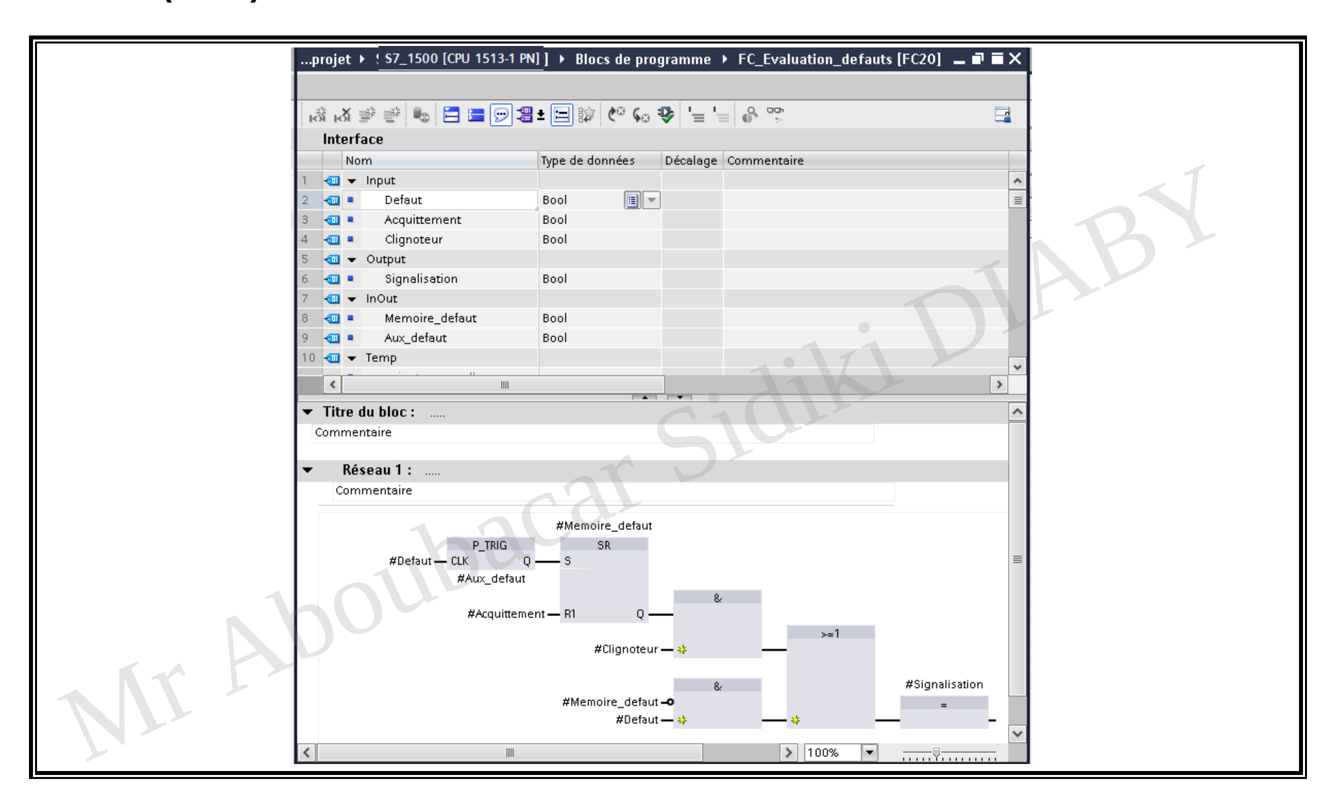
Proposition de résolution

Une réponse au front du signal de défaut est nécessaire car si le défaut perdure, la « mémoire de défaut » est immédiatement remise à 1 après l'acquittement. Dans ce cas, la LED de signalisation du défaut se remettrait à clignoter

Si la « mémoire défaut » est mise à 1 (le message n'a pas encore été acquitté), l'opération combinatoire ET supérieure entraîne le clignotement du voyant.

L'opération combinatoire ET inférieure entraîne l'allumage continu du voyant lumineux si le défaut perdure après acquittement.

9.6.2. Exercice 1 : Editer une fonction paramétrable « FC_Evaluation_defauts » (FC20)



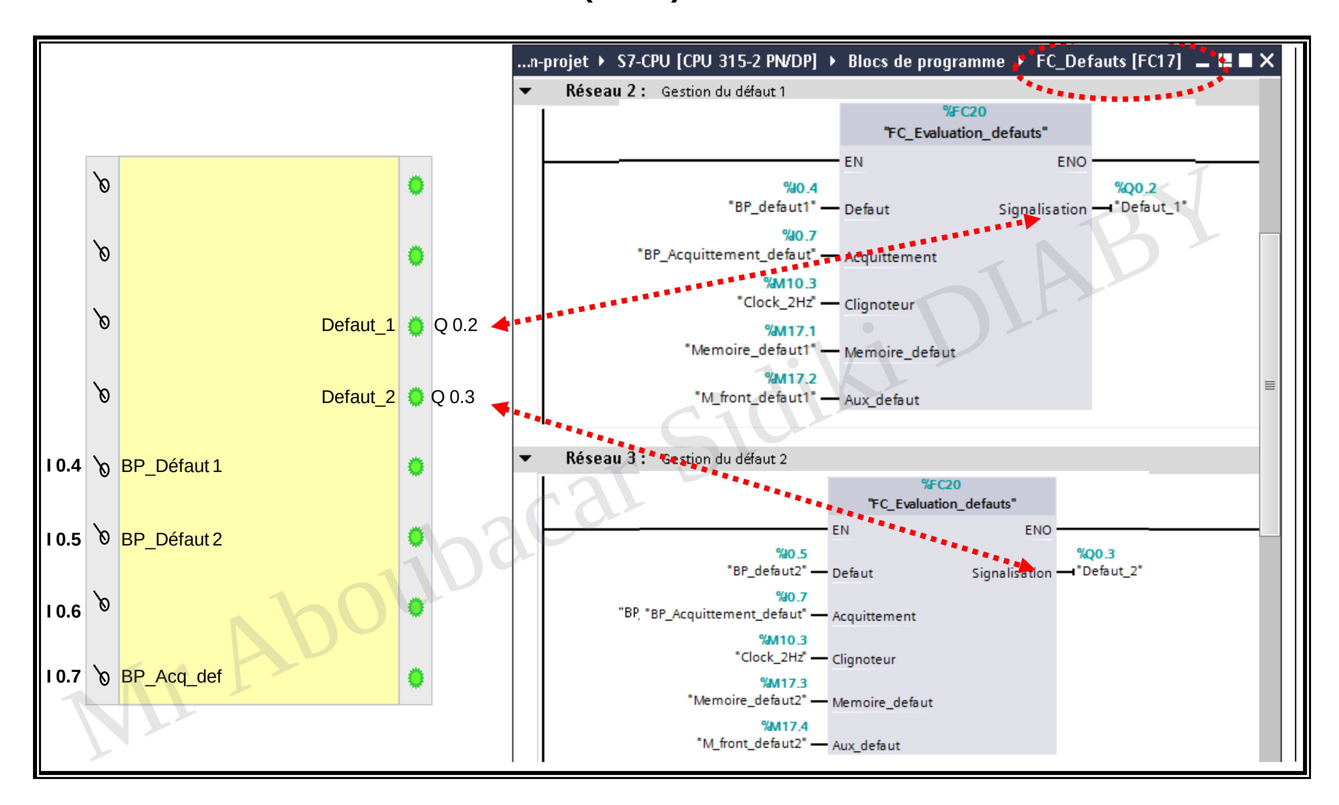
Énoncé

Créez le programme d'évaluation des défauts dans le bloc paramétrable (fonction) FC 20 « Evaluation_defauts ».

Marche à suivre

1. Insérez le bloc « FC_Evaluation_defauts » (FC20) dans le dossier « Blocs de programme ».
2. Déclarez les opérandes formels comme indiqué ci-dessus.
3. Créez le programme comme indiqué ci-dessus.
4. Enregistrez le bloc.

9.6.3. Exercice 2 : Appeler et affecter la fonction paramétrable « FC_Evaluation_defauts » (FC20)



Fonction actuelle

Dans le bloc « FC_Defauts » (FC 17), la surveillance du temps de transport des pièces est programmée.

Énoncé

Outre la surveillance du temps de transport des pièces, 2 signaux de défaut provenant du processus (signal des deux commutateurs du simulateur) doivent être évalués. Pour cela, il faut appeler deux fois le bloc « FC_Evaluation_defauts » (FC20) qui a déjà été programmé.

Marche à suivre

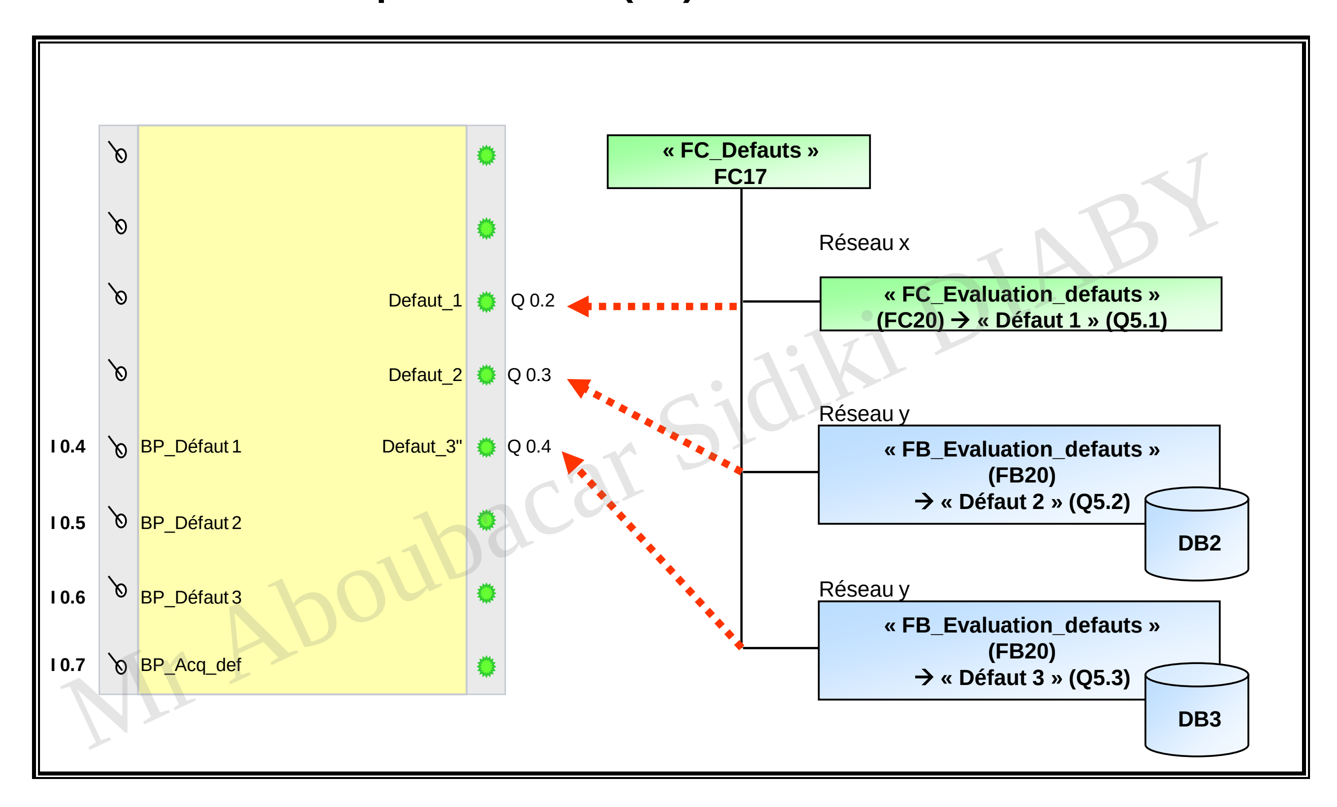
1. Programmez les deux appels du nouveau bloc FC 20 dans le bloc FC 17, comme indiqué ci-dessus.
2. Enregistrez le bloc FC 17 modifié et chargez-le dans la CPU.
3. Testez votre programme pour vérifier les fonctions décrites.

Nota

L'octet de mémento MB10 est déjà paramétré en tant qu'octet de mémento de cadence.

Le mémento « Clock_2Hz » (M10.3) a une fréquence de clignotement de 2Hz et a déjà été créé comme variable API.

9.7. Énoncé : Evaluation des signaux de défaut avec un bloc fonctionnel paramétrable (FB)

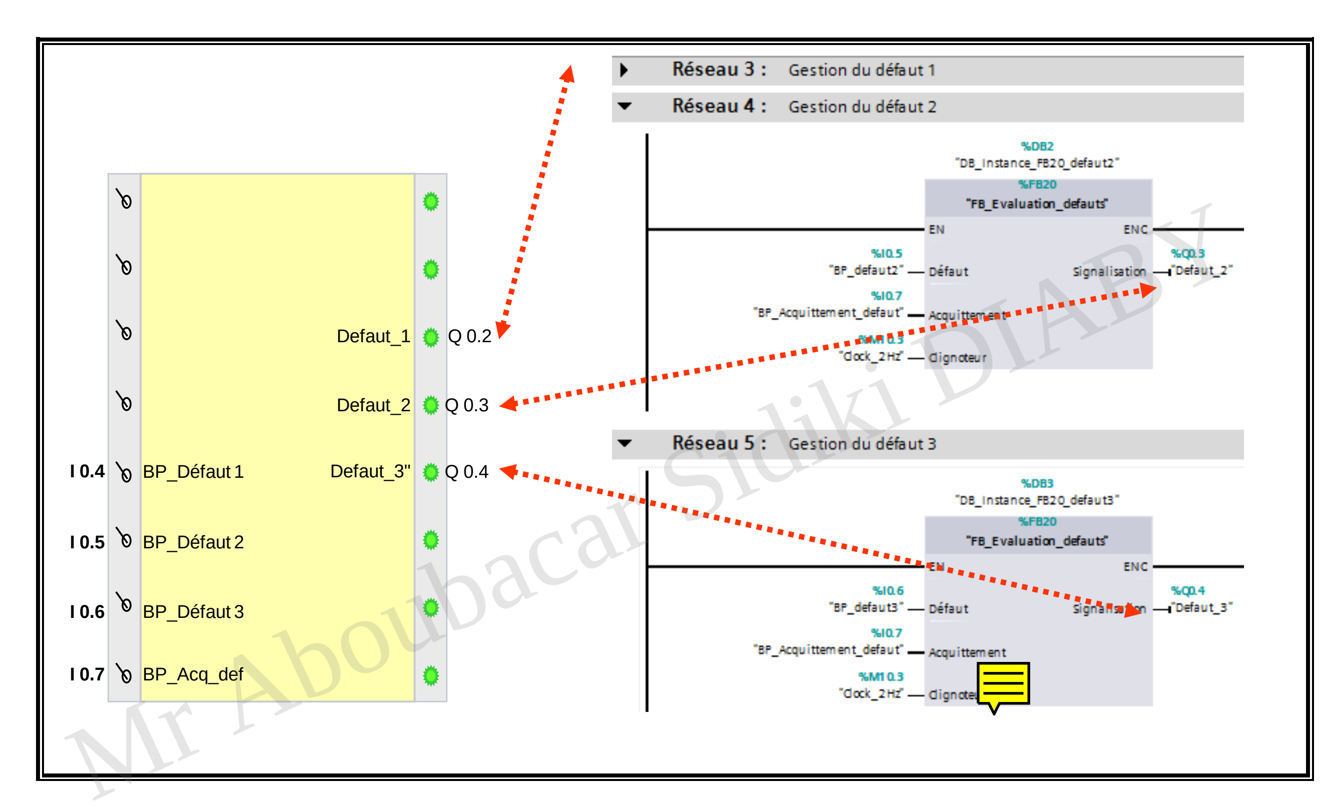


Énoncé

L'évaluation des signaux de défaut décrite ci-avant doit désormais être réalisée avec un bloc fonctionnel (FB), et non plus avec une fonction (FC). Avantage : pour l'évaluation des fronts du signal de défaut requise en interne, il n'est pas nécessaire de transmettre au FB des variables globales de l'extérieur en tant que mémoire de signalisation, le FB pouvant utiliser des variables locales statiques.

Le signal de défaut 1 doit, comme auparavant, être évalué par le bloc existant « FC_evaluation_defauts » (FC20), mais l'évaluation des signaux de défaut 2 et 3 doit être assurée par le nouveau bloc fonctionnel « FB_evaluation_defauts » (FB20).

9.7.1. Instanciation des FB



Particularités

A la différence des fonctions (FC), les blocs fonctionnels (FB) ont une mémoire. Cela veut dire qu'un bloc de données, appelé bloc de données d'instance, est associé au bloc fonctionnel. Lors de l'appel d'un FB, il faut aussi indiquer le numéro du DB d'instance afin que celui-ci soit ouvert automatiquement.

Le DB d'instance permet de mémoriser les variables statiques. Ces variables locales peuvent être utilisées uniquement dans le FB dans la partie déclarative duquel elles ont été déclarées, mais elles sont conservées après fermeture du bloc.

Paramètres

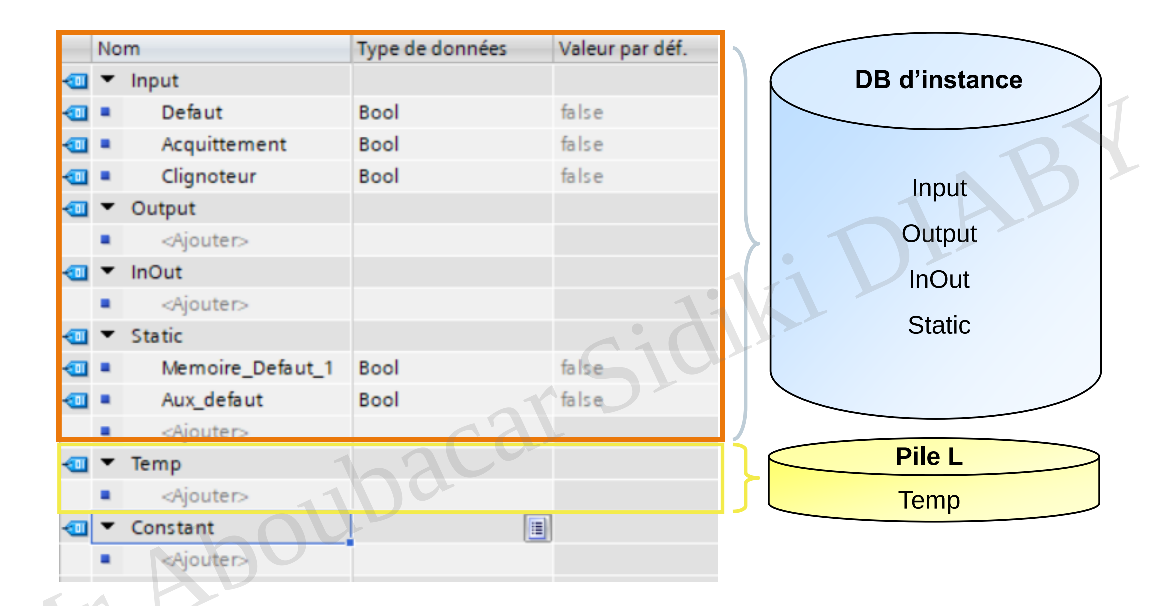
L'appel d'un bloc fonctionnel entraîne le stockage des valeurs des opérandes actuels dans le bloc de données d'instance. Si aucun opérande actuel n'a été attribué à l'opérande formel lors de l'appel du bloc, la dernière valeur stockée dans le DB d'instance pour ce paramètre est utilisée lors du traitement du programme.

Chaque appel du FB peut donner lieu au transfert d'autres opérandes actuels. Les données sont conservées dans le bloc de données après fermeture du bloc fonctionnel.

Avantages des FB

- Pour la programmation du bloc FC, l'utilisateur doit rechercher et gérer lui-même des zones de mémentos ou de données libres, alors que les variables statiques d'un FB sont gérées par le logiciel STEP 7.
- Le risque de double affectation des zones de mémentos ou de données, inhérent à la programmation du bloc FC, disparaît avec l'utilisation de variables statiques.
- Les variables statiques « Memoire_defaut » et « aux_defaut » remplacent dans le FB les opérandes formels de même nom du FC20. L'appel de bloc est simplifié du fait de la disparition des deux paramètres.

9.7.2. Partie déclarative du bloc fonctionnel



Paramètres

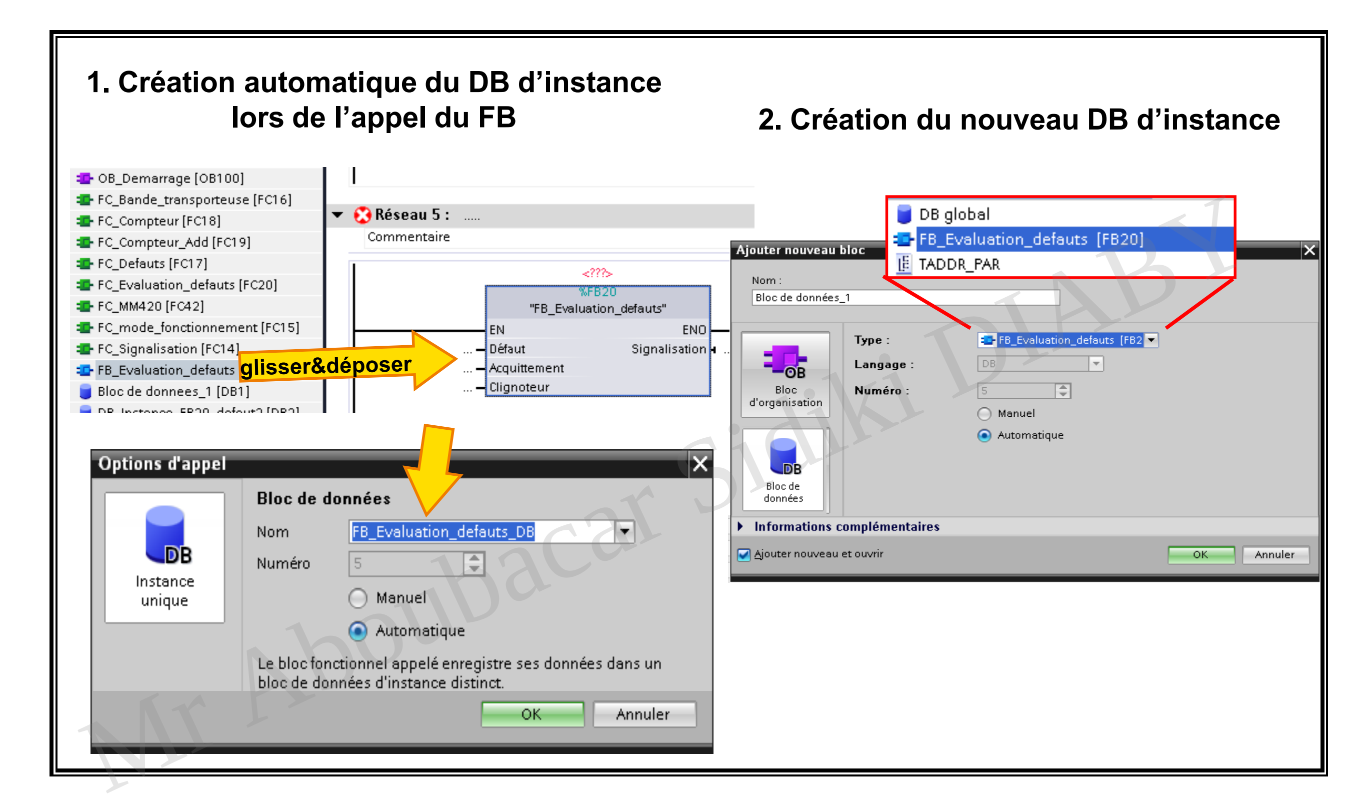
L'appel d'un bloc fonctionnel entraîne le stockage des valeurs des opérandes actuels dans le bloc de données d'instance. Si aucun opérande actuel n'a été attribué à l'opérande formel lors de l'appel du bloc, la dernière valeur stockée dans le DB d'instance pour ce paramètre est utilisée lors du traitement du programme.

Chaque appel du FB peut donner lieu au transfert d'autres opérandes actuels. Les données sont conservées dans le bloc de données après fermeture du bloc fonctionnel.

Variables statiques

Contrairement aux fonctions, les blocs fonctionnels disposent de « variables statiques » (Static). Ces variables constituent la mémoire du FB car elles ne sont pas stockées dans la pile locale, mais dans une zone mémoire propre, appelée DB d'instance.

9.7.3. Création de blocs de données d'instance



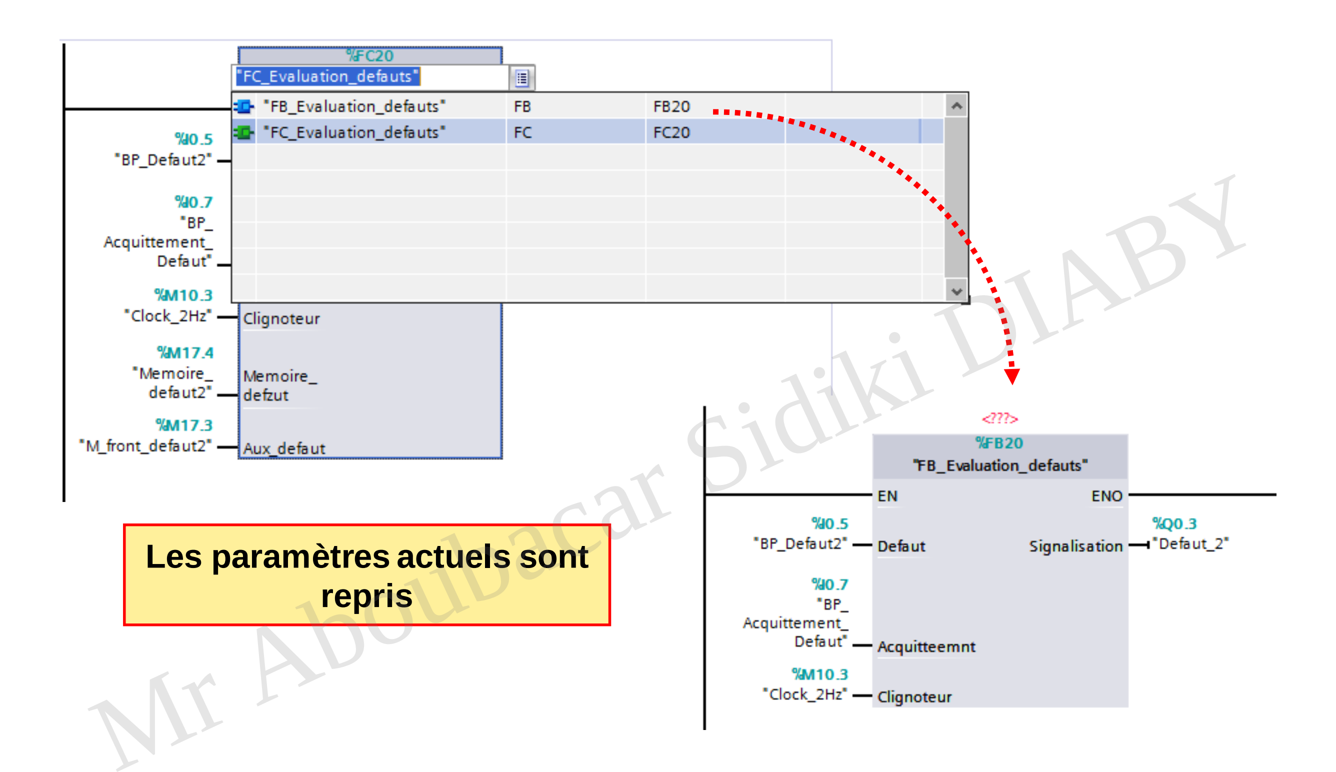
Création d'un DB d'instance
Il existe deux manières de créer un nouveau DB d'instance :

- Lors de l'appel du FB, l'utilisateur indique le DB d'instance qui sera affecté au FB. Dans la boîte de dialogue, l'utilisateur peut indiquer le nom symbolique du bloc et, le cas échéant, un numéro manuel du DB d'instance.
- Lors de la déclaration d'un nouveau DB, on choisit le type « Bloc fonctionnel XY ».


Un DB d'instance ne peut être associé qu'à un seul FB. Mais à chaque appel d'un FB, celui-ci peut se voir affecter un autre DB d'instance.


Attention !
Si le FB est modifié (ajout de paramètres ou de variables statiques supplémentaires), il faut ensuite générer à nouveau le DB d'instance.

9.8. Changement l'appel d'un bloc

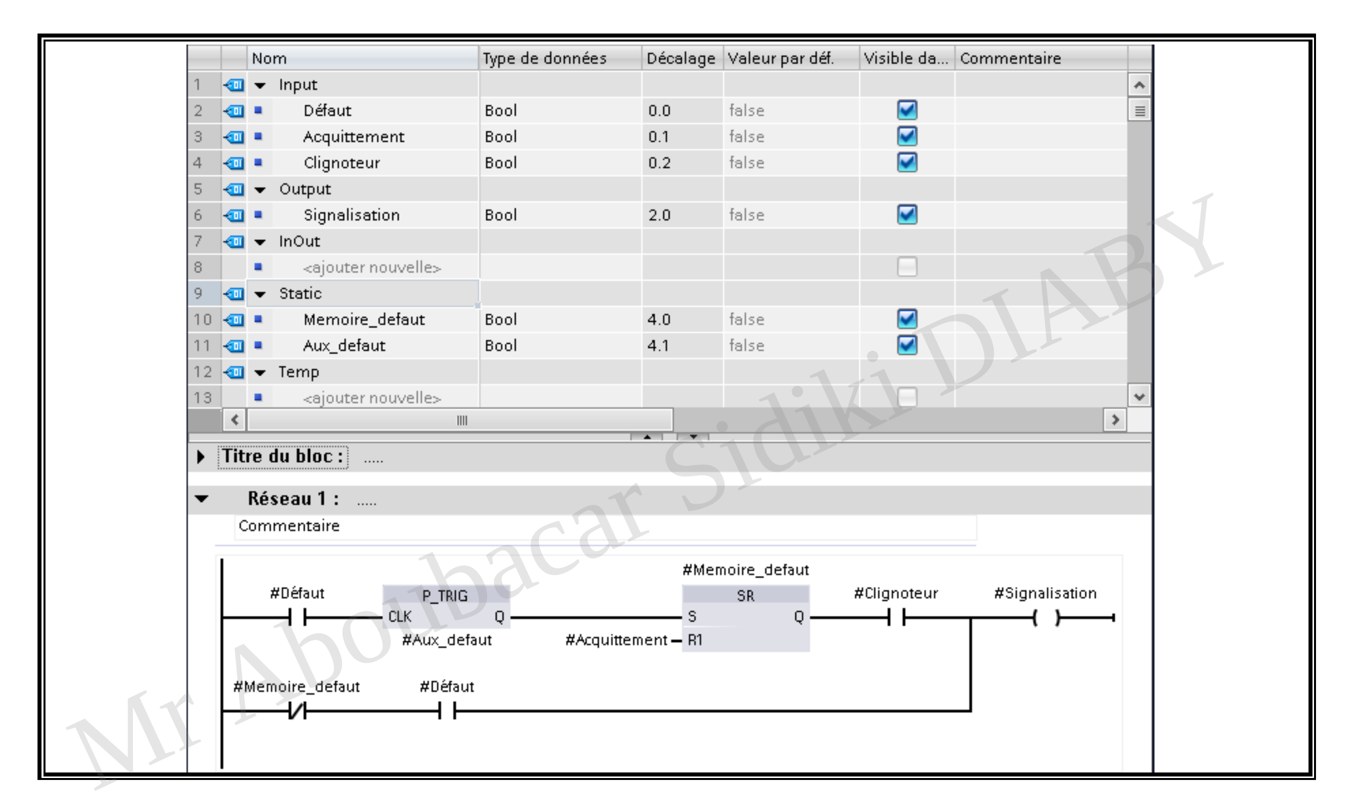


Afin de remplacer l'appel d'un bloc par un autre appel de bloc, une liste déroulante de tous les FBs et FCs peut-être ouverte au point d'appel en double-cliquant sur le nom du bloc déjà appelé.

Avantage :

Si les deux blocs ont les mêmes paramètres formels, alors ils conservent leurs paramètres réels et tous les paramètres formels n'ont pas besoin d'être fournis avec de nouveaux paramètres réels.

9.9. Exercice 3 : Editer le bloc fonctionnel « FB_Evaluation_defauts »



	Nom	Type de données	Décalage	Valeur par déf.	Visible da...	Commentaire
1	Input					
2	Défaut	Bool	0.0	false	☑	
3	Acquittement	Bool	0.1	false	☑	
4	Clignoteur	Bool	0.2	false	☑	
5	Output					
6	Signalisation	Bool	2.0	false	☑	
7	InOut					
8	<ajouter nouvelle>				☐	
9	Static					
10	Memoire_defaut	Bool	4.0	false	☑	
11	Aux_defaut	Bool	4.1	false	☑	
12	Temp					
13	<ajouter nouvelle>				☐	

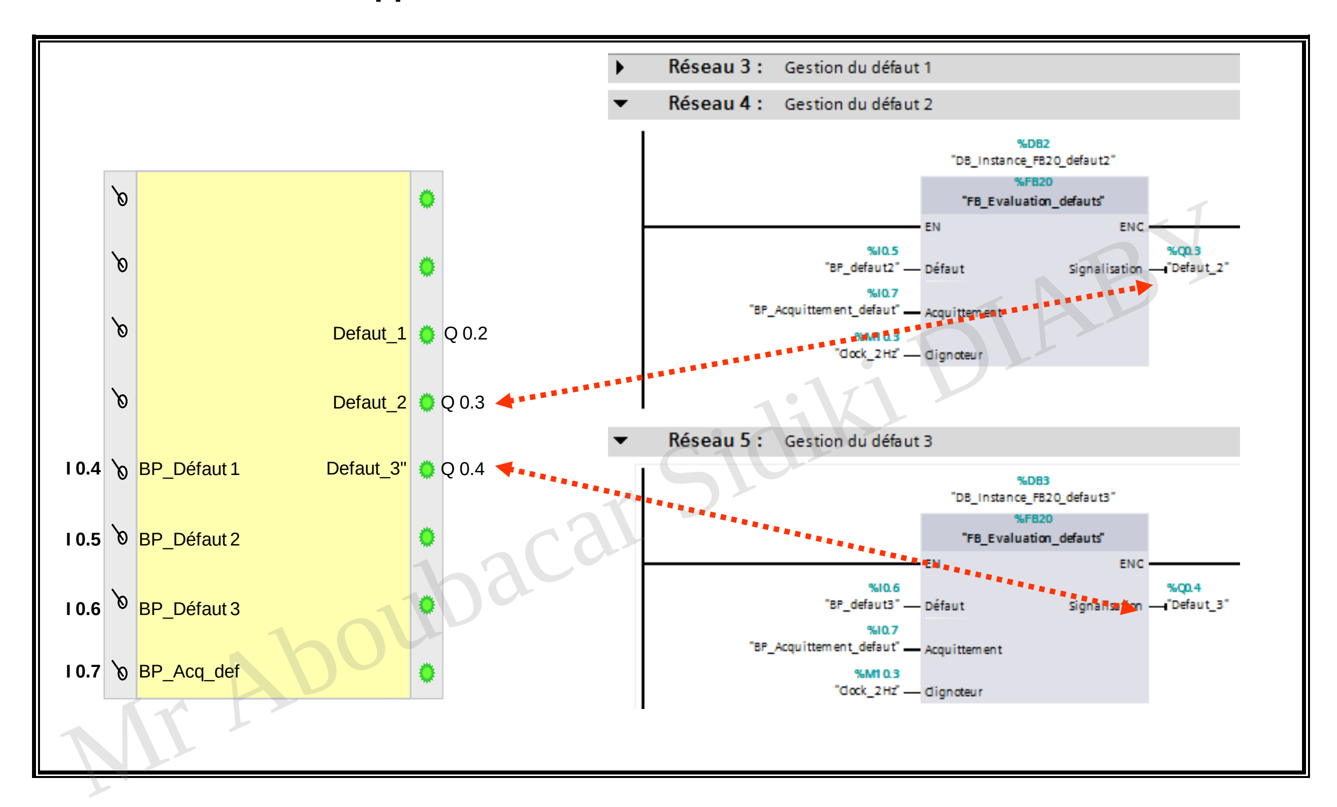
Énoncé

Créez le nouveau bloc « FB_Evaluation_defauts » pour évaluer les défauts 2 et 3.

Marche à suivre

1. Insérez le nouveau bloc « FB_Evaluation_defauts » (FB20).
2. Déclarez les opérandes formels et les variables statiques du bloc comme indiqué ci-dessus.
3. Programmez le FB 20 en copiant les séquences de programme requises du FC 20 déjà programmé.
4. Enregistrez le bloc et chargez-le dans la CPU.

9.9.1. Exercice 4 : Appel du bloc fonctionnel « FB_Evaluation_defauts »



Énoncé

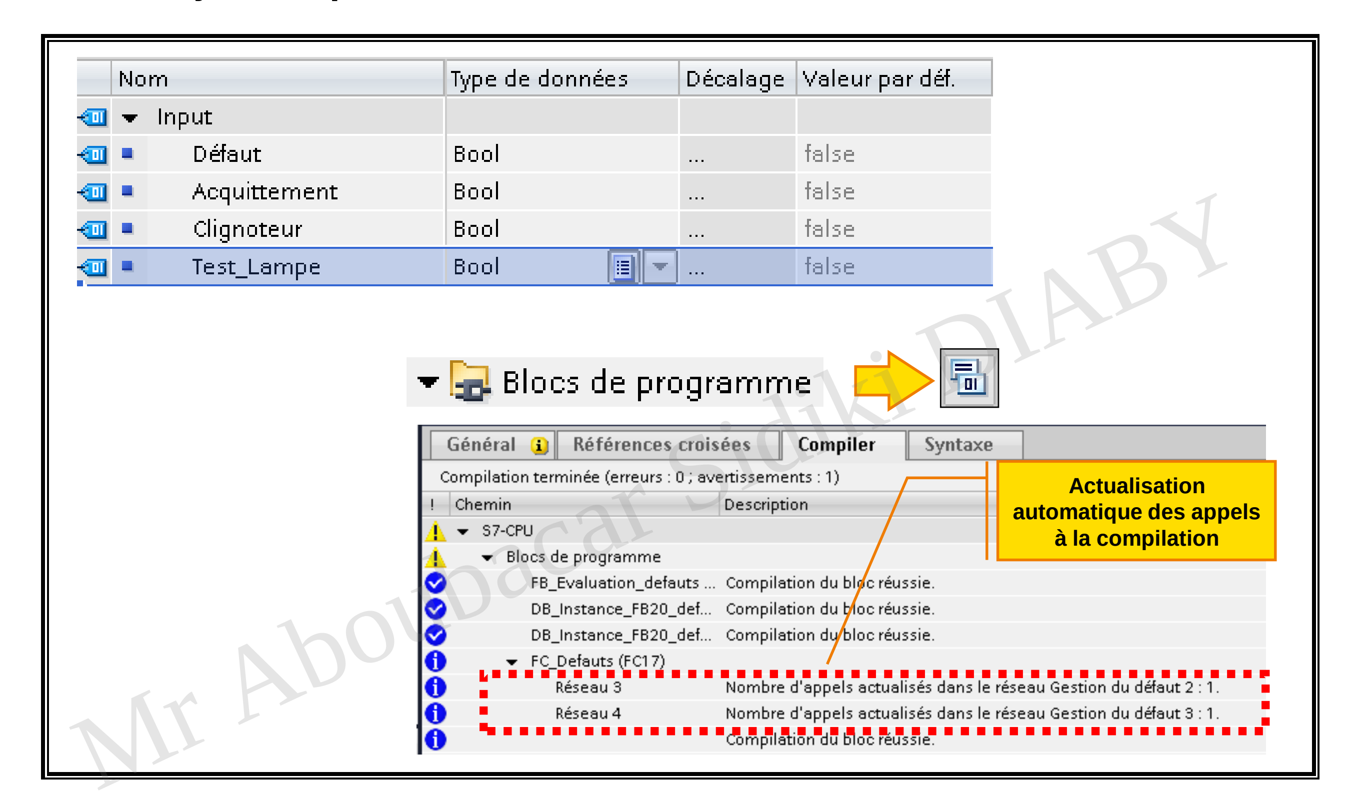
L'analyse du défaut 2 (programmée jusqu'à présent par l'appel du bloc FC 20) et l'analyse du nouveau défaut 3 doivent être réalisées avec le nouveau FB 20.

Le bloc paramétrable FB 20 doit pour cela être appelé deux fois avec, pour chaque appel, un bloc de données d'instance différent dans le bloc FC 17.

Marche à suivre

1. Effacez dans le FC 17 le 2e appel du bloc FC 20 car l'évaluation du défaut 2 sera désormais réalisée avec le FB 20.
2. Programmez les deux appels du FB 20 comme indiqué ci-dessus dans deux nouveaux réseaux dans le FC 17 et générez avec l'éditeur les DB d'instance 2 et 3.
3. Enregistrez le FC 17 modifié.
4. Chargez l'ensemble du programme dans la CPU et vérifiez le bon fonctionnement
5. Enregistrez votre projet.

9.10. Ajout de paramètres de bloc



Problématique

L'adaptation ou la modification d'interfaces ou du code dans certains blocs pendant ou après la création du programme peut provoquer des conflits d'horodatage qui risquent d'occasionner à leur tour des incohérences entre les blocs appelants et les blocs appelés (ou les blocs de référence), nécessitant alors d'importantes corrections.

Si des paramètres de bloc sont **ajoutés** ultérieurement, dans un bloc déjà appelé dans le programme, il faut aussi actualiser les appels de ce bloc dans les autres blocs.

- Actualisation manuelle

  Dans le bloc appelant ouvert, les appels de bloc incohérents sont repérés en rouge. En cliquant sur les appels incohérents avec le bouton droit de la souris, on peut sélectionner la fonction « Actualiser». Apparaît alors une fenêtre qui permet de visualiser l'ancien appel (erroné) et le nouvel appel de bloc (dans l'illustration ci-dessus, avec le paramètre supplémentaire « Test_Lampe »). Pour les blocs fonctionnels, le DB d'instance correspondant est ensuite généré à nouveau.

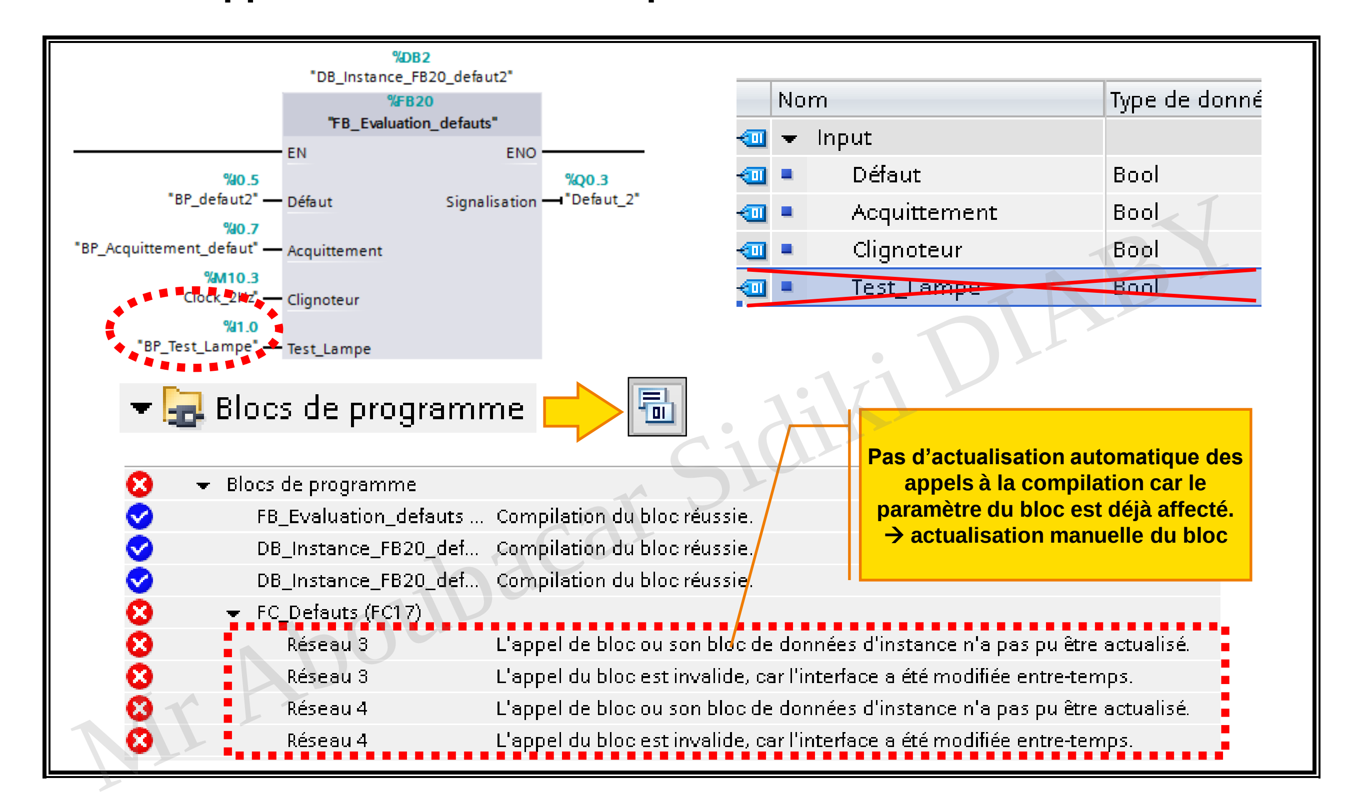
- Actualisation automatique

  Les éventuels conflits d'horodatage sont détectés lors de la compilation du programme utilisateur complet. En cas d'ajout de paramètres, les appels de blocs concernés sont automatiquement actualisés.

  Pour les fonctions, le paramètre formel ajouté doit être affecté dans le programme avant le chargement dans la CPU (affectation obligatoire des paramètres).

  Pour les blocs fonctionnels, si un paramètre formel n'est pas affecté, le programme utilise la valeur par défaut dans le DB d'instance correspondant (affectation par défaut des paramètres).

9.11. Suppression ultérieure de paramètres de bloc



Problématique

L'adaptation ou la modification d'interfaces ou du code dans certains blocs pendant ou après la création du programme peut provoquer des conflits d'horodatage qui risquent d'occasionner à leur tour des incohérences entre les blocs appelants et les blocs appelés (ou les blocs de référence), nécessitant alors d'importantes corrections.

Si des paramètres de bloc sont **supprimés** ultérieurement, dans un bloc déjà appelé dans le programme, il faut aussi actualiser les appels de ce bloc dans les autres blocs.

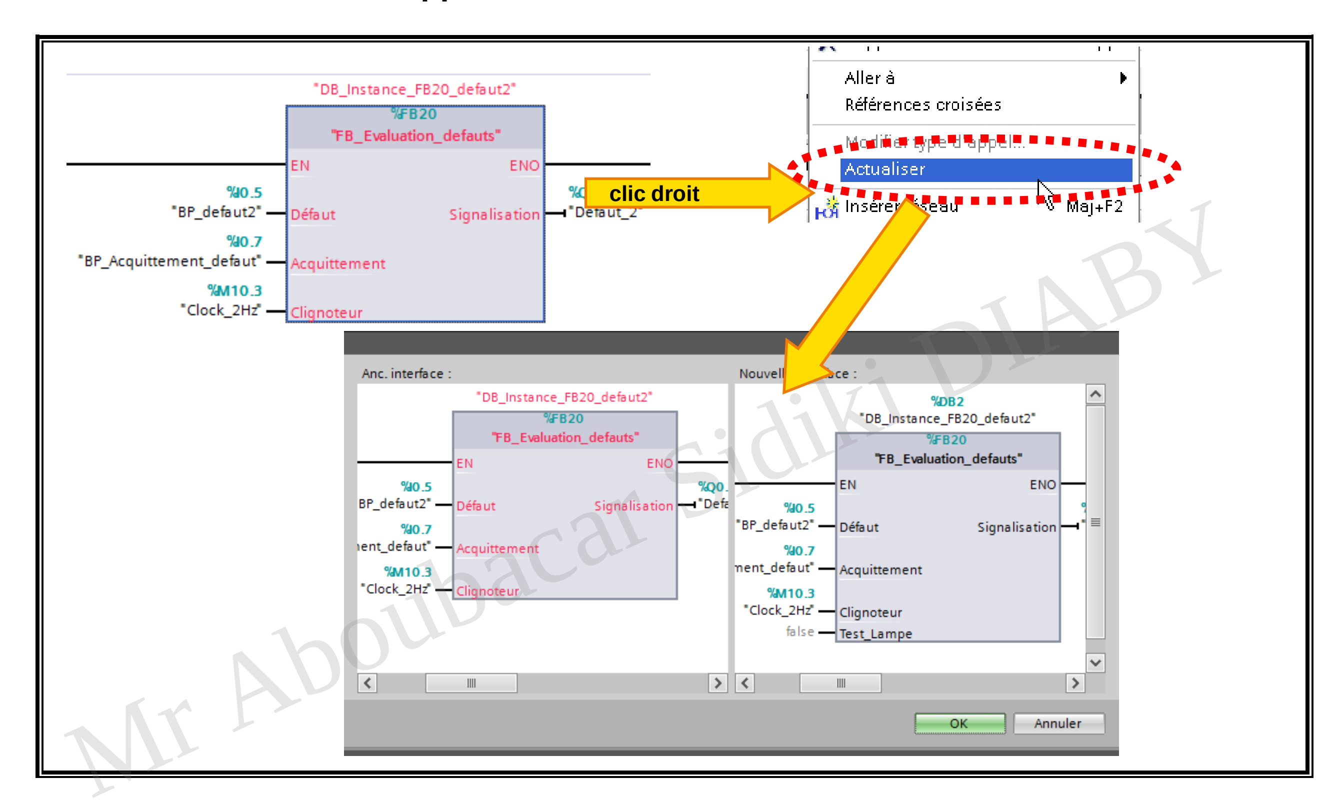
- Actualisation manuelle

  Dans le bloc appelant ouvert, les appels de bloc incohérents sont repérés en rouge. En cliquant sur les appels incohérents avec le bouton droit de la souris, on peut sélectionner la fonction « Actualiser». Apparaît alors une fenêtre qui permet de visualiser l'ancien appel (erroné) et le nouvel appel de bloc (dans l'illustration ci-dessus, avec le paramètre supplémentaire « Test_Lampe »). Pour les blocs fonctionnels, le DB d'instance correspondant est ensuite généré à nouveau.



L'actualisation automatique des appels de blocs n'est pas gérée car les paramètres formels supprimés sont en général affectés à des opérandes actuels dans le bloc appelant et que le logiciel ne supprime jamais des séquences de programme lors de la génération des blocs.

9.11.1. Correction de l'appel de bloc



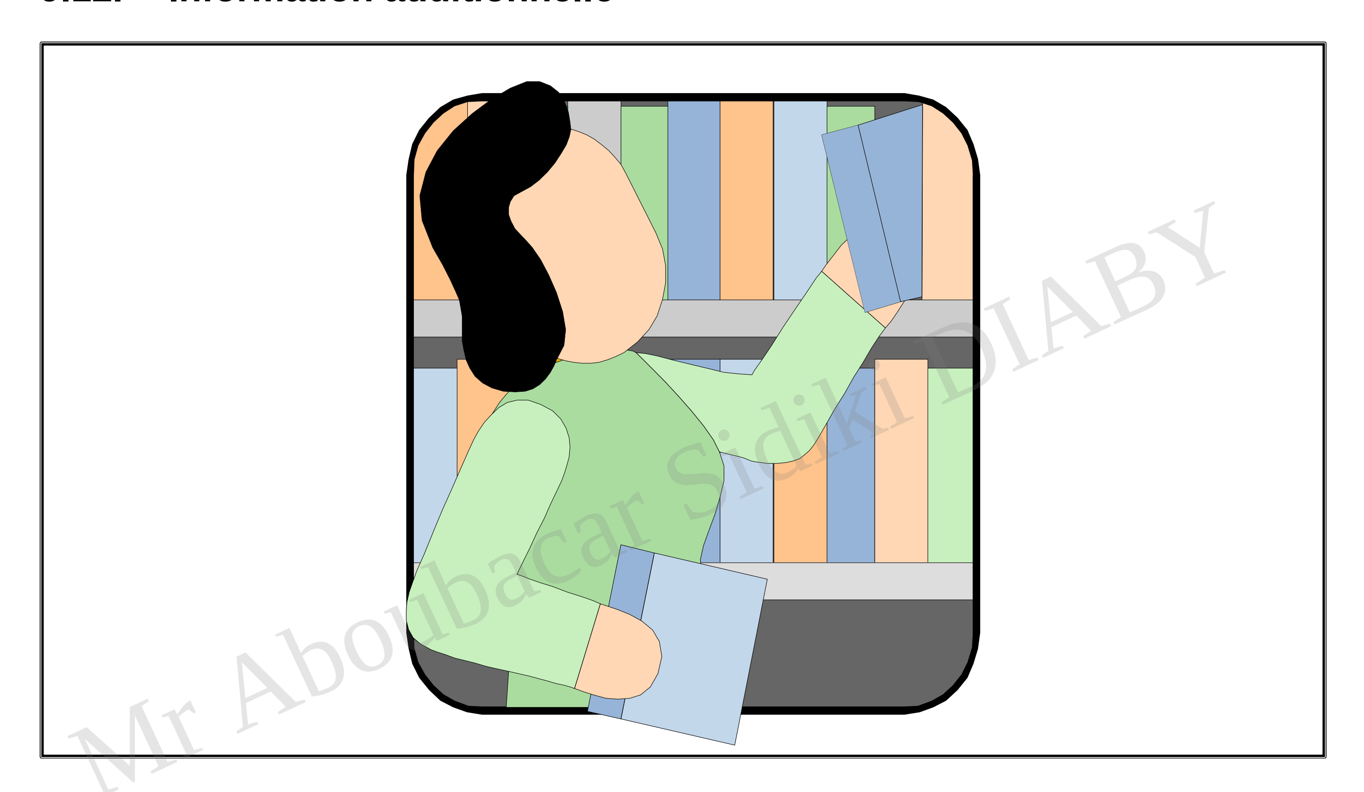
Actualisation manuelle

Les appels de bloc incohérents sont repérés en rouge dans le bloc appelant ouvert. Cliquez avec le bouton droit de la souris sur l'appel de bloc incohérent et sélectionnez l'option de menu « Actualiser ». Dans la fenêtre affichée apparaissent l'ancien appel de bloc (erroné) et le nouvel appel de bloc (dans notre exemple avec le paramètre supplémentaire « Test_Lampe »). Ensuite, pour les blocs fonctionnels, le DB d'instance est généré à nouveau.



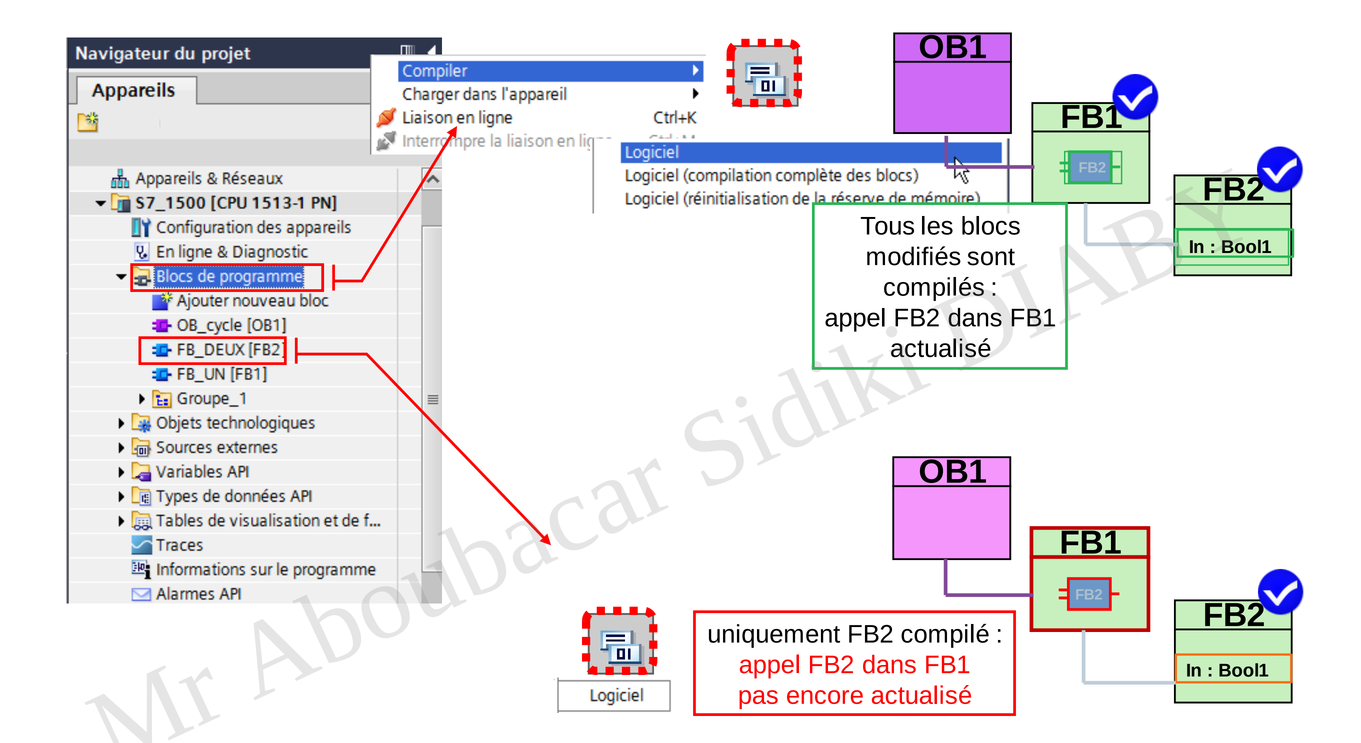
Les blocs de données d'instance nouvellement générés ne sont pas automatiquement chargés dans la CPU. C'est à l'utilisateur de le faire explicitement.

9.12. Information additionnelle

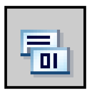




9.12.1. Compilation de blocs individuels/tous les bocs modifiés



Compile → Software (Changement seulement):



Si un bloc individuel est sélectionné dans l'arborescence du projet ou si une compilation est déclenchée par le bouton affiché à gauche lorsqu'un bloc est ouvert, seul ce bloc unique est compilé (s'il a été modifié). L'inconvénient de la compilation d'un bloc individuel est que les conflits d'interface dans les blocs d'appel provoqués par les changements d'interface ne sont pas corrigés.

Si plusieurs blocs sont sélectionnés ou un groupe de blocs, seuls les blocs modifiés depuis la dernière compilation sont compilés (compilation delta).

Si le dossier "Blocs de programme" est sélectionné, la compilation delta pour l'ensemble du programme est effectuée.

Compile → Software (Reconstruire tous les blocs / ou Tout reconstruire) :

Tous les blocs, même ceux qui ne sont pas modifié depuis la dernière compilation, sont compilés.

Compile → Réinitialiser la mémoire:

Remarque : La réserve de mémoire est traitée dans programmation 2.
Avec le processus de compilation, la réserve de mémoire des blocs de données est également réinitialisée, c'est-à-dire que les variables créées ultérieurement au cours des extensions de blocs de données sont supprimées de la réserve de mémoire et intégrées dans la partie régulière du bloc de données.

Résultats des compilations

L'état de la compilation est affiché hiérarchiquement dans la fenêtre Inspecteur "Info -> Compiler". Si des erreurs se sont produites lors de la compilation, vous pouvez accéder directement à l'emplacement de l'erreur en double-cliquant sur l'entrée d'erreur.

9.12.2. Variables locales et globales

	Variables globales	Variables locales
Domaine de validité	• valables dans l'ensemble de la CPU et utilisables par tous les blocs. • le nom des variables doit être univoque dans l'ensemble de la CPU.	• valables uniquement dans le bloc dans lequel elles sont déclarées • le nom de la variable doit être univoque au sein du bloc
Opérandes	• Entrées • Sorties • Mémentos • Variables dans blocs de données • Temporisations/compteurs SIMATIC	• Variables temporaires (dans tous les blocs de code) • Variables statiques (uniquement dans FB)
Lieu de la déclaration	• Table de variables API • Blocs de données globaux	• Partie déclaration du bloc
Représentation	• Représentation entre des guillemets Exemple : “Max“	• Représentation avec un # Exemple : #Max

Plage de validité des variables

Les variables déclarées dans la table de variables automate ou dans un bloc de données global peuvent être adressées par tous les blocs de programme CPU. Pour cette raison, ces variables sont appelées variables globales.

Les variables et les paramètres déclarés dans la partie de déclaration d'un bloc de code (logique) sont des opérandes locaux; ils ne peuvent être utilisés que dans la partie déclaration du même bloc.

Sommaire 10

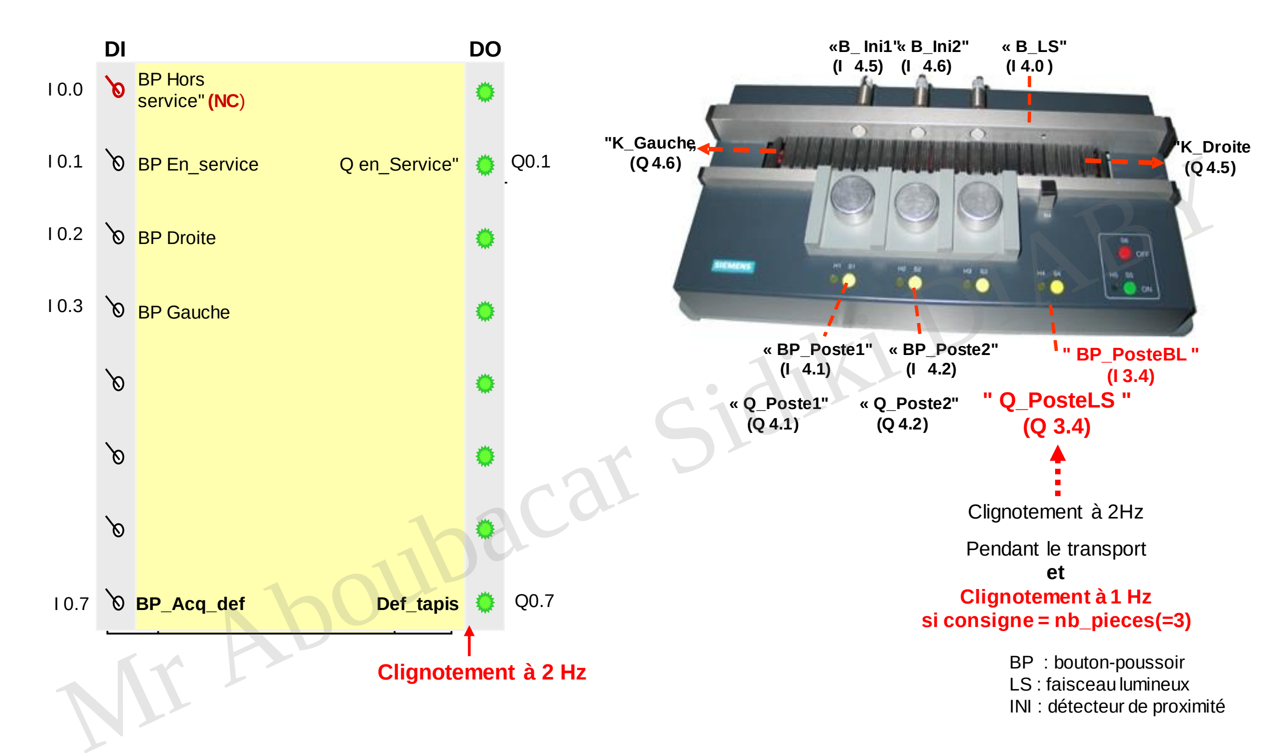
10. Opérations numériques

A l'issue du chapitre, vous allez…



- … **connaître différents types de données**
- … **connaître les fonctions mathématiques**
- … **connaître les fonctions de comparaison**
- … **connaître la fonction MOVE pour l'affectation de valeurs**
- … **connaître différentes fonctions de comptage et de temporisation**
- … **savoir utiliser et programmer des fonctions de temporisation et savoir sélectionner le type de données approprié**

10.1. Enoncé : Comptage des pièces transportées



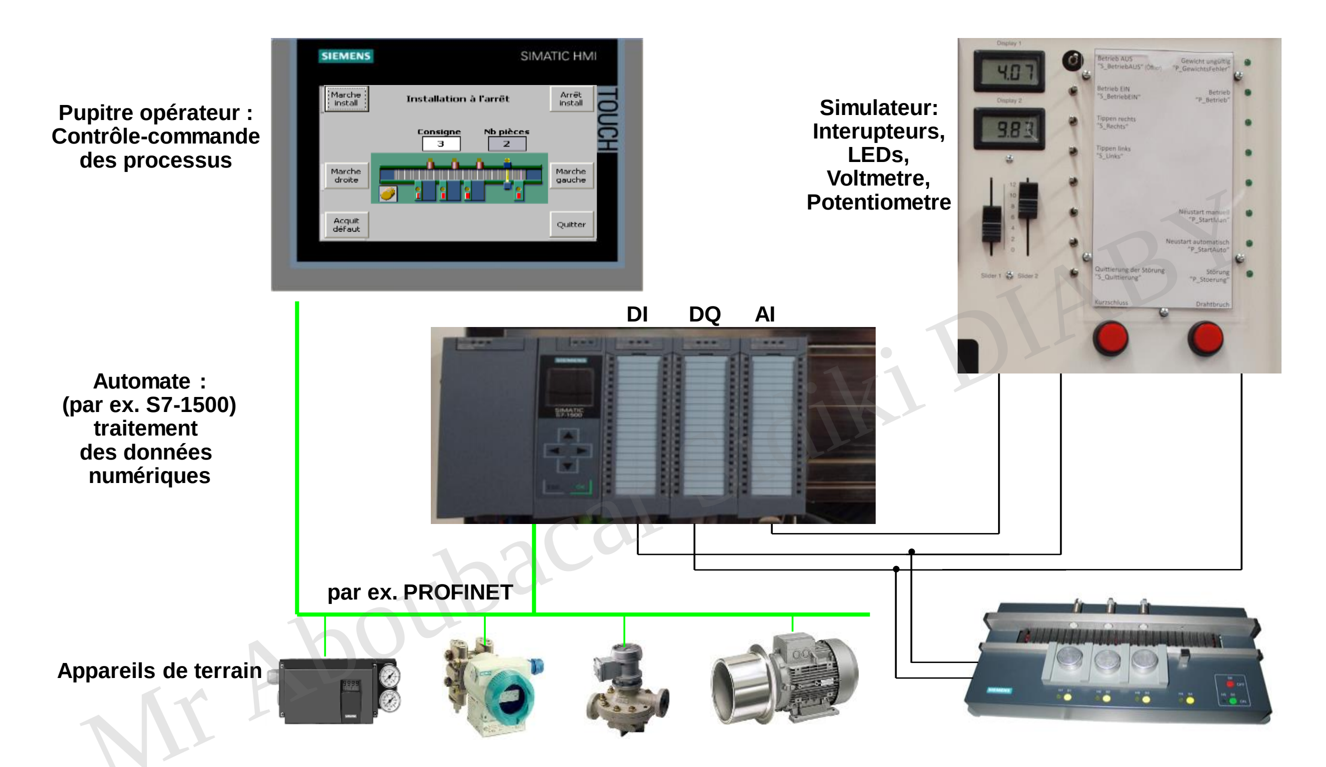
Fonction existante

Lorsque « Q_En_Service » est activé (Q0.1), les pièces sont transportées du poste 1 ou 2 jusqu'au poste de contrôle final, c'est-à-dire jusqu'au franchissement de la barrière lumineuse. L'opération de transport commence dès qu'une pièce est déposée au poste 1 ou 2 et que le bouton correspondant est actionné. Elle s'arrête dès que la pièce a franchi la barrière lumineuse..

Enoncé

- Lorsque « Q_En_Service » (Q0.1) est activé, les pièces transportées doivent être comptées dès qu'elles franchissent la barrière lumineuse (I 3.0) (« B_LS » 0->1).
- Le compteur doit être remis à zéro à l'activation de « Q_En_Service » (Q0.1). Le compteur peut-être réinitialisé avec le bouton poussoir « BP_PosteBL » (I 3.4)..
- Le voyant "Q_PosteLS" (Q3.4) clignote à 1Hz quand le nombre de pièce compté atteint la consigne. Aucune pièce ne doit être placé sur le convoyeur (bloqué dans « FC_Signalisation » FC14) et il est impossible de démarrer une autre séquence de transport (bloqué dans « FC_Convoyeur FC16). Le voyant doit clignoter à 2Hz pendant le transport des pièces.
- En outre, le temps requis pour le transport d'une pièce de la baie 1 et de la baie 2 jusqu'à la barrière lumineuse est mesuré.
- Si le temps de transport est supérieur à 6 secondes, le convoyeur est arrêté et il est indiqué à la sortie "Def_Tapis" (Q0.7) avec un clignotement de 1Hz.
- Une fois que cette erreur a été acquittée avec « BP_Acq_defaut » (I 0.7), une nouvelle pièce peut être transportée.

10.1.1. Acquisition, traitement et sortie des données



Traitement binaire/numérique

Faisant appel à une logique purement combinatoire, les automates traitent exclusivement des données numériques binaires. Leur rôle dans le traitement des données numériques a peu à peu été renforcé par l'augmentation des performances des calculateurs de commande, la complexité croissante des tâches à réaliser et les exigences de qualité. Les données numériques se rencontrent dans tous les domaines de la commande, aussi bien au niveau de la surveillance et du pilotage des processus qu'au niveau de la commande des appareils de terrain.

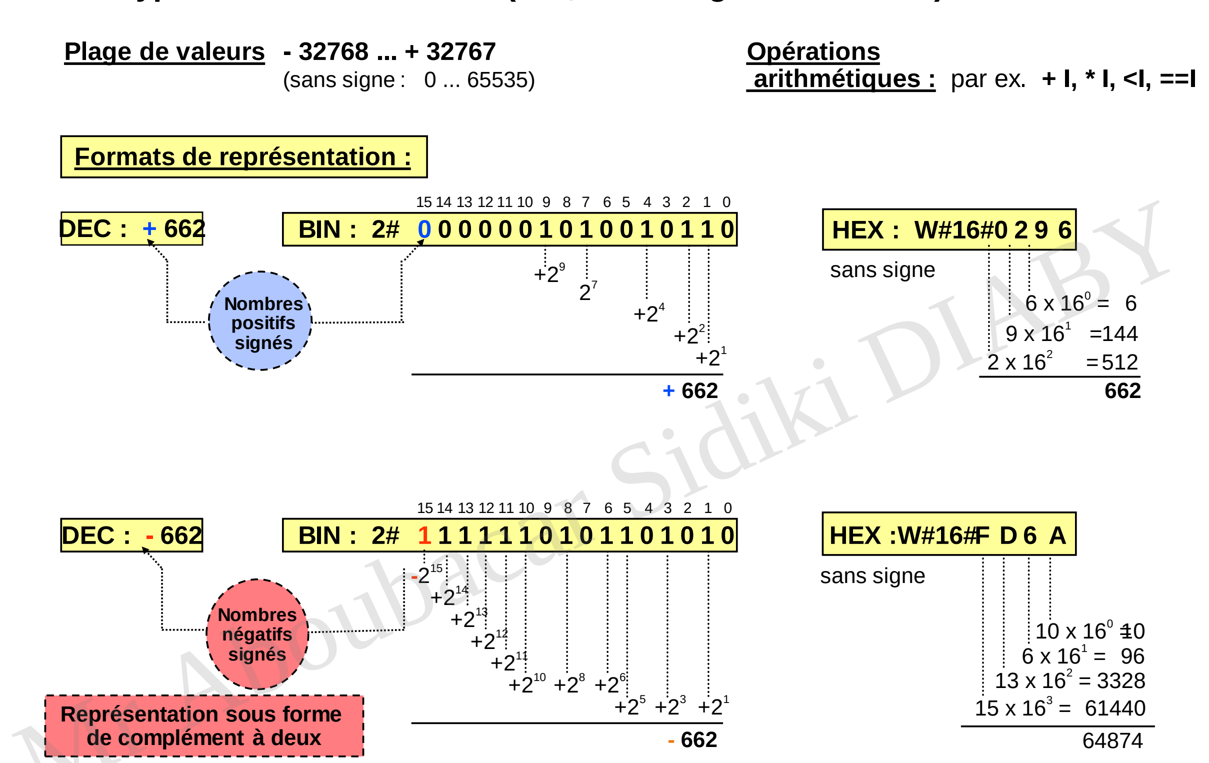
Contrôle-commande

La surveillance de processus a pour but de fournir à l'utilisateur des informations instantanées et claires sur le déroulement du processus et de lui permettre d'intervenir dans la commande. Si l'on utilisait autrefois des appareils d'entrée et sortie dépourvus de logique, comme les afficheurs numériques 7 segments ou les roues codeuses, on fait appel aujourd'hui à des terminaux de contrôle-commande « intelligents » raccordés à un automate programmable. Pour la transmission des données entre terminal et automate ou encore pour la mémorisation et le traitement des données au sein de l'automate, on utilise différents formats de codage numérique des données.

Appareils de terrain

Les appareils de terrain chargés d'acquérir les données du processus ou d'intervenir dans son déroulement sont, eux aussi, alimentés directement en données numériques via des bus de terrain. Le raccordement d'appareils de terrain, comme les entraînements ou les systèmes de pesage, via des modules d'entrée/sortie analogiques appartient de plus en plus au passé.

10.1.2. Type de données : Entier (INT, entier signé sur 16 bits)



Type de données Integer (entier sur 16 bits)

Le type de données Integer (INT) correspond à un nombre entier, c'est-à-dire sans chiffres après la virgule. SIMATIC S7 enregistre les valeurs du type de données Integer sous forme de nombres signés codés sur 16 bits, ce qui donne la plage de valeurs indiquée ci-dessus. SIMATIC S7 offre à l'utilisateur des opérations arithmétiques pour le traitement des nombres entiers.

Décimal

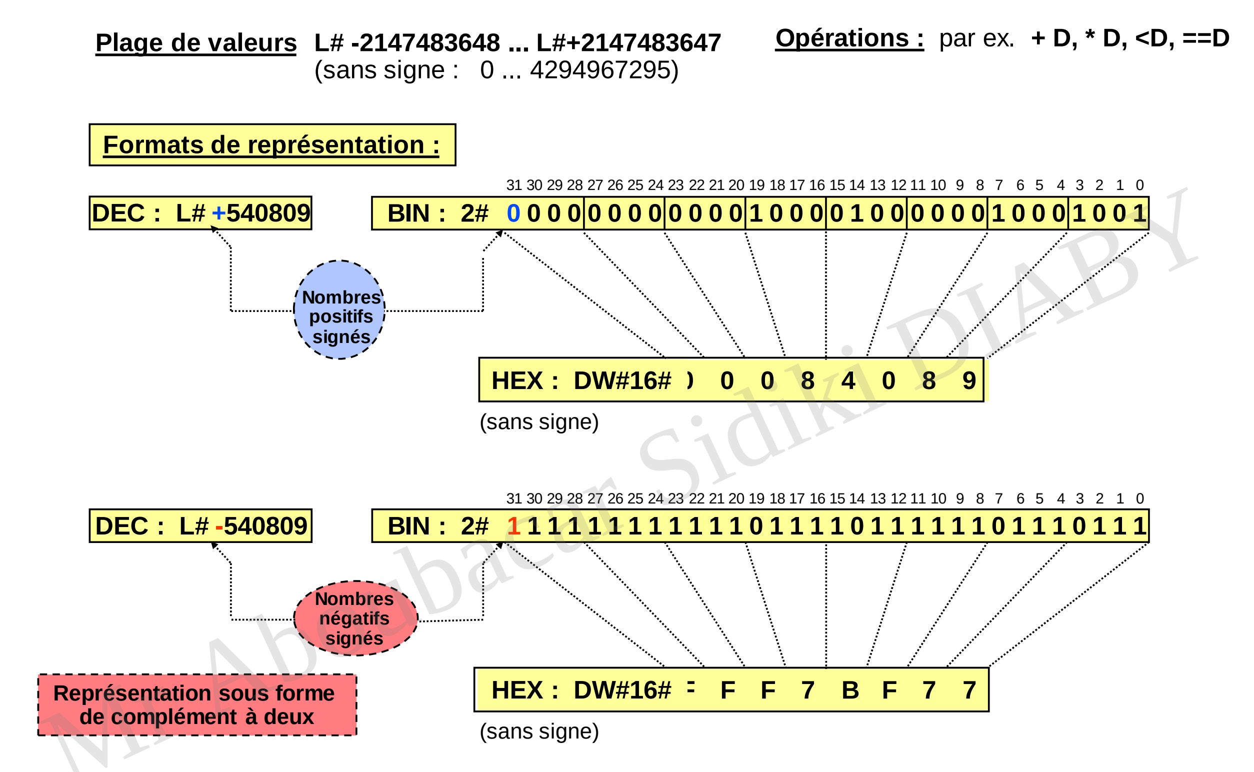
Dans STEP7, les constantes du type de données Integer sont entrées sous forme décimale (non BCD) avec signe et sans indication explicite du format. Si l'entrée d'entiers constants sous forme *binaire, octale* et *hexadécimale* est par principe possible, elle est souvent inappropriée car difficile à lire. C'est la raison pour laquelle la syntaxe de STEP7 ne permet de définir les entiers que sous une forme décimale.

Binaire

Dans un calculateur numérique, toutes les valeurs sont enregistrées sous forme codée binaire. Le système de numération binaire ne connaît que les chiffres 0 et 1. La base 2 utilisée par ce système résulte de ce nombre de chiffres disponibles. Le poids de chaque chiffre d'un nombre binaire correspond donc à une puissance de la base 2. Cette base s'exprime sous la forme 2#... dans l'indication du format.

En format binaire, les valeurs négatives sont représentées en complément à deux. Dans ce cas, le bit de plus fort poids (bit 15 pour le type de données Integer) prend la valeur - 215. Cette valeur étant supérieure à la somme des autres valeurs, ce bit contient aussi l'indication du signe. S'il est égal à 0, la valeur est positive, s'il est égal à 1, elle est négative. La conversion d'un nombre binaire en nombre décimal s'effectue en additionnant les poids des bits à « 1 » (voir figure).

10.1.2.1. Type de données : Entier double (DINT, entier signé sur 32 bits)



Double integer (entier sur 32 bits)

SIMATIC S7 enregistre les valeurs du type de données *Double integer (DINT)* sous forme de nombres signés codés sur 32 bits. La plage de valeur va de -2147483648 à +2147483648.

Hexadecimal

Le système de nombre hexadécimal fournit 16 chiffres différents (0 à 9 et A à F). Cela résulte en Base 16 de ce système de nombres. Par conséquent, la valeur de chaque bit d'un nombre hexadécimal résulte d'une puissance de base 16. Les nombres hexadécimaux sont spécifiés avec 16 # pour identifier le système de numérotation de base. Le nombre de bits spécifiables est variable de 1 à 16. Les chiffres A à F correspondent aux valeurs décimales 10 à 15. La valeur 15 est la dernière valeur pouvant être codée en binaire - sans signe - avec 4 bits. Hors de cette corrélation, la conversion simple d'un nombre binaire en un nombre hexadécimal et vice versa peut être obtenue. De cette manière, quatre bits binaires peuvent chacun composer facilement un chiffre d'un nombre hexadécimal.

Nombre octal

Le système de nombre octal fournit 8 chiffres différents (0 à 7). Cela résulte en Base 8 de ce système de nombres. En conséquence, la valeur de chaque bit d'un nombre octal résulte d'une puissance de base 8. Les nombres octaux sont spécifiés avec 8 # pour identifier le système de numérotation de base. La valeur 7 est la valeur qui peut être codée en binaire - sans signe - avec 3 bits. De cette manière, trois bits binaires peuvent chacun composer un chiffre d'un nombre octal.

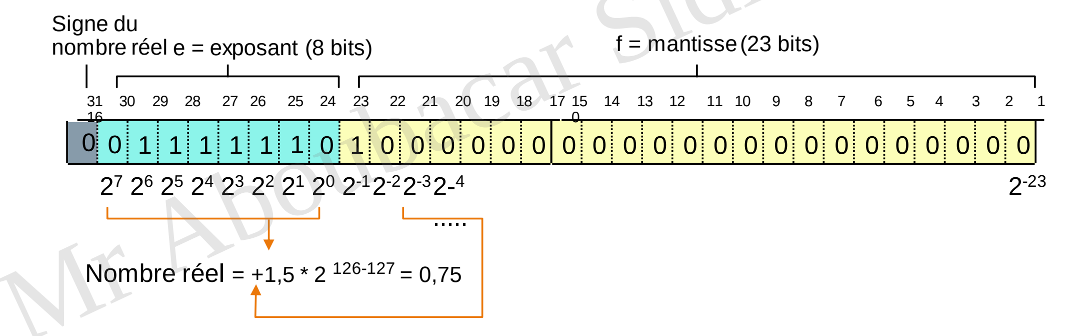
10.1.3. Type de données : Réel (REAL & LREAL, nombre à virgule flottante sur 32 bits)

Plage de valeurs **- 3.402823•10^{+38} ... - 1.175495•10^{-38} , 0.0, + 1.175495•10^{-38} … + 3.402823•10^{+38}**

Opérations : par ex. **+ R, * R, <R, ==R**
sin, acos, ln, exp, SQR

Format général d'un nombre réel = (signe) • (1.f) • (2^{e-127})

Exemple : 0.75 (Lors de la saisie sur la PG, on utilise un point à la place de la virgule)



Real/LReal

Les types de données INT et DINT précédemment décrits s'utilisent pour enregistrer des valeurs entières signées. Ces valeurs ne permettent donc d'effectuer que des opérations dont le résultat est aussi un nombre entier. Cependant, lorsque les valeurs traitées sont des grandeurs analogiques de type courant, tension ou température, l'utilisation de nombres au format Real est indispensable (nombres à virgule flottante). Pour pouvoir représenter ces valeurs, il faut définir des positions binaires dont la valeur est inférieure à 1 (puissances de la base 2 avec exposant négatif).

Format

Pour pouvoir représenter la plus grande plage de valeurs possible (voir figure) avec une place mémoire définie (SIMATIC S7 : double mot, 32 bits), il est nécessaire de pouvoir choisir l'emplacement de la virgule selon ses besoins. Pour faciliter le traitement d'une position de virgule variable, l'IEEE a défini très tôt un format pour les nombres à virgule flottante, qui a été ensuite repris par la norme IEC 61131 et, par conséquent, par STEP 7. Le code binaire d'un nombre à virgule flottante est constitué d'un bit de signe, d'un exposant (8 bits) et d'une mantisse. Les valeurs réelles sont définies sans indication de format. Lors de l'entrée d'une valeur réelle constante (exemple : 0.75), l'éditeur procède à un formatage automatique (exemple : 7.5000e-001).

Application

Les nombres à virgule flottante s'utilisent notamment pour le traitement des valeurs analogiques. Leur principal avantage réside dans le fait qu'ils admettent une multitude d'opérations, parmi lesquelles des opérations standard comme +, -, * , / mais aussi des opérations comme sin, cos, exp, ln, ..., qui trouvent surtout leur application dans les algorithmes de régulation.

10.1.4. Types de données et formats d'affichage

Nom	Adresse	Format d'affichage	Valeur visualisation	Valeur d...	⚡	Commentaire
	%IW0	DEC	1			
	%IW0	Bin	2#0000_0000_0000_0001			
	%I0.7	Bool	FALSE			
"BP-Marche"	%I0.0	Bool	FALSE			
"Acquittement_defaut"	%I1.0	Bool	TRUE			
"Roue_codeuse"	%IW2	Hexa	16#1234			
"Roue_codeuse"	%IW2	Bin	2#0001_0010_0011_0100			
"Roue_codeuse"	%IW2	DEC+/-	4660			
	%MW20	DEC+/-	1234	1234	☑ ⚠	
	%MW20	Hexa	16#04D2			
	%MW20	Bin	2#0000_0100_1101_0010			
	%MB20	Hexa	16#04			Octet de poids fort
	%MB21	Hexa	16#D2			Octet de poids faible
	%MD80	DEC	1234	1234	☑ ⚠	Type de données DINT
	%MD80	Hexa	16#0000_04D2			
ATTENTION chevauchement !!!	%MW83	Hexa	16#D244			
	%MD84	Nombre à virgule...	1234.0	1234.0	☑ ⚠	
	%MD84	Hexa	16#449A_4000			

Formats d'affichage

Les fonctions test « Visualisation/forçage de variables » et « Visualiser bloc » offrent différents formats d'affichage du contenu des variables ou des registres. Chaque variable peut en principe être visualisée dans n'importe quel format d'affichage, mais il est préférable de choisir le format le mieux adapté au type de données.

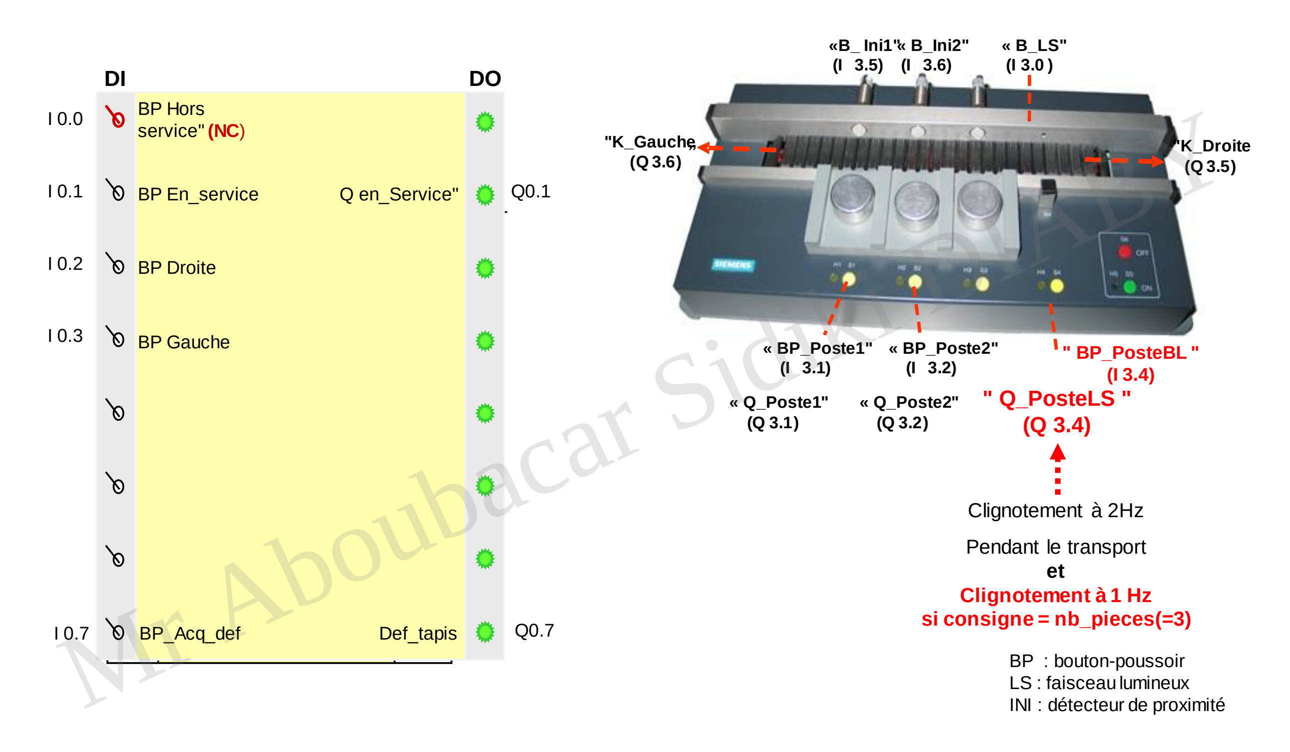
- **BOOL** : Affichage d'un bit individuel (possible uniquement avec les variables de type BOOL)
- **BIN** : Affichage des différents bits d'une variable (intéressant pour les variables de type BYTE, WORD, DWORD)
- **HEX,BCD** : Affichage du contenu d'une variable sous forme de nombre hexadécimal, ou un nombre BCD (intéressant pour les variables de type BYTE, WORD, DWORD, LWORD)
- **DEC** : Affichage du contenu d'une variable sous forme de nombre décimal (non BCD) signé (intéressant pour les variables de type USINT, UINT, UDINT, ULINT)
- **DEC +/-** : Affichage du contenu d'une variable sous forme de nombre décimal signé (non BCD) (intéressant pour les variables de type SINT, INT, DINT, LINT)
- **Virgule flottante** : Affichage du contenu d'une variable sous forme de nombre à virgule flottante (intéressant pour les variables de type REAL, LREAL)
- ***et autres…***

Adressage

La mémoire de SIMATIC S7 est entièrement orientée octet. Le mot de mémento MW 20 contient par ex. les octets de mémento MB 20 (octet de poids fort) et MB 21 (octet de poids faible, voir figure), le double mot de mémento MD 80, les octets de mémento MB 80, 81, 82 et 83.

En cas d'accès absolu à des variables (par ex. avec MD 82), il faut veiller à ce que le format de l'accès (MD... dans ce cas) et l'adresse (toujours égale à l'adresse de l'octet de poids fort, 82 dans ce cas) soient corrects. Dans le cas contraire, une valeur non valide serait chargée (par ex. avec MW 83, voir figure). L'adressage symbolique de variables (« MD_DINT » dans ce cas) permet d'éviter ces erreurs.

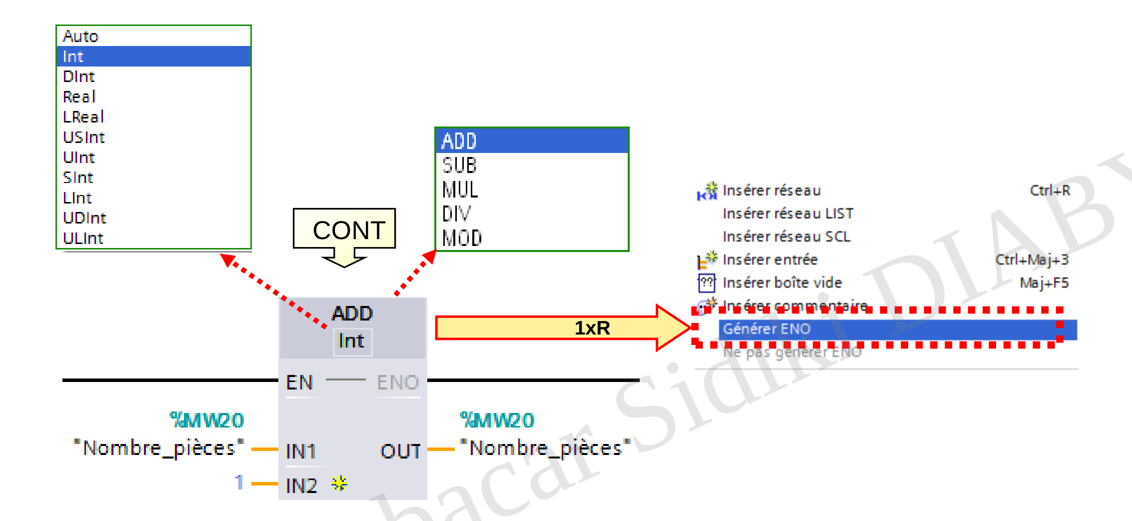
10.2. Enoncé: Comptage des pièces transportées en utilisant l'addition avec le bloc « FB_Comptage_Add »



Enoncé

- Quand "Q_En_Service"(Q0.1) est activé, les pièces transportées sont comptés dès qu'elles franchissent la barrière lumineuse (I3.0) (« B_LS » 0 ->1).
- Quand la consigne de pièce est atteinte, le voyant « Q_PosteLS » (Q3.4) s'allume et aucun transport supplémentaire n'est possible.
- Tant qu'aucun transport n'est possible, les voyants des postes 1 et 2 sont éteint et signale par la même occasion que le transport n'est pas possible.
- Le compteur doit être remis à zéro à l'activation de « Q_En_Service » (Q0.1). Le compteur peut-être réinitialisé à tout moment avec le bouton poussoir « BP_PosteBL » (I 3.4)

10.2.1. Fonctions mathématiques : Addition



Opérations arithmétiques

Il existe une série d'opérations arithmétiques disponibles pour le traitement des variables des types de données arithmétiques, tels qu'entier (INT), double entier (DINT) et réel (REAL).

Entrées et Sorties des éléments LAD/FBD :

- **EN (activer)**

L'exécution de l'opération peut-être déterminée comme suit via l'entrée EN :

- EN n'est pas connecté: L'opération est toujours exécutée (indépendamment du RLO)
- L'opération logique EN est remplie (RLO = 1): L'opération est exécutée.
- L'opération logique EN n'est pas remplie (RLO = 0): L'opération n'est pas exécutée.

- **IN1 / IN2**

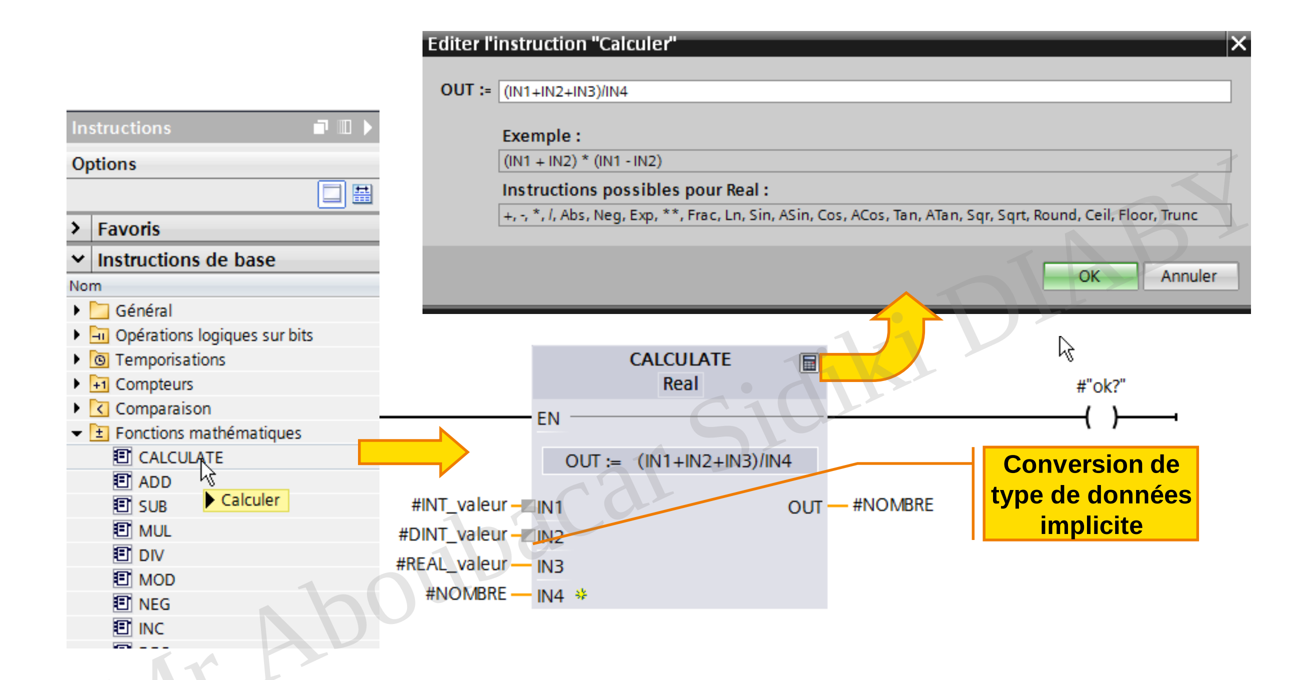
Le calcul arithmétique est appliqué aux valeurs founies à IN1 et IN2 et le résultat es envoyé à OUT.

- **ENO (sortie activable)**
(Vous pouvez active et désactiver la generation des sorties ENO via la menu contextuelle. Attention : le paramètre est valide pour toutes les prochaines fonctions qui seront insérés dans le projet)

SI la fonction ENO est activée, il peut accepter les valeurs suivantes :

- ENO = 0 puisque le parameter reel à l'entrée EN a la valeur FAUX
  - La sortie n'es pas écrit (la variable fournie à la sortie n'est pas écrite, c'est-à-dire qu'ele conserve sa valeur d'origine)
- ENO = 0 parce que une erreur est survenue.
  - OUT contains an invalid value (the variable delivered to OUT is overwritten with an invalid value)
  - La sortie contient une valeur non valide (la variable fournie à la sortie est remplacée par une valeur non valide)
- ENO = 1 (instruction a été executer sans erreur):
  - La sortie contient le résultat (la variable fournie à la sortie est remplacé par le résultat)

10.2.1.1. Boite CALCULATE

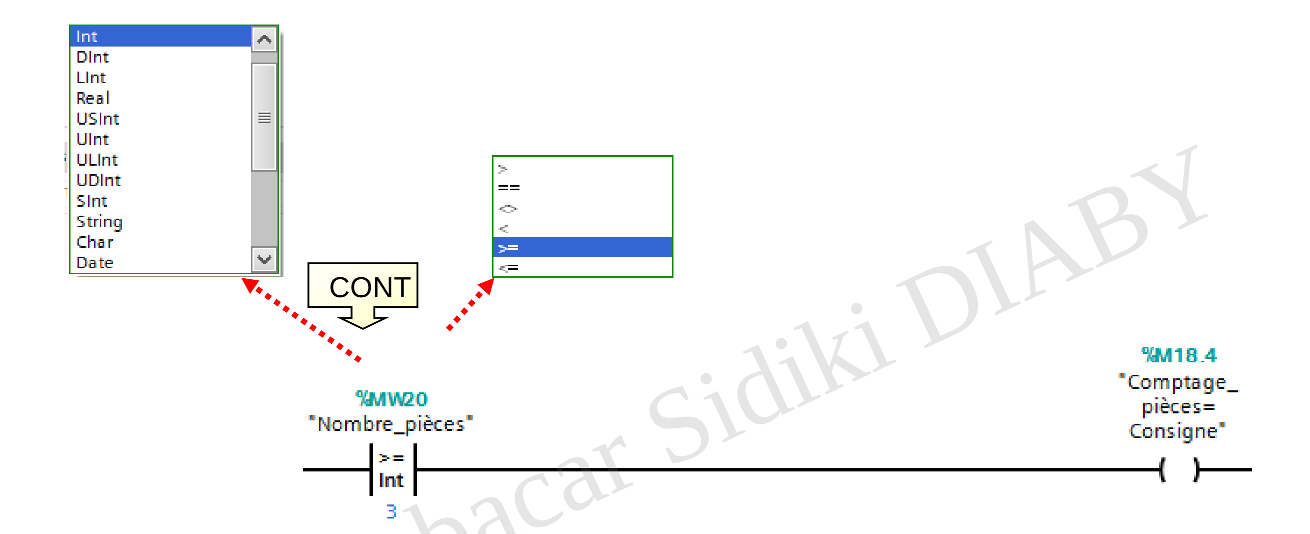


Boite de calcul

Avec la boîte de calcul, vous pouvez combiner des calculs qui contiennent plusieurs opérations mathématiques différentes

La conversion de type de données implicite est disponible aux entrées et aux sorties de la boîte et une vérification peut être faite à travers la sortie ENO pour savoir si des erreurs sont survenues dans le calcul, comme des débordements dans les conversions de type ou les opérations mathématiques, ou le résultat de la boîte de calcul est sans erreur.

10.2.2. Fonctions de comparaison



Fonction de comparaison

Avec les instructions de comparaison, les paires de valeurs numériques suivantes peuvent-être comparées :

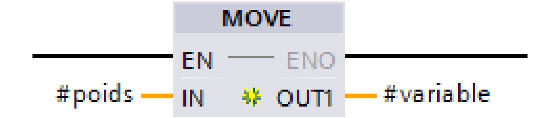
- Toutes les variations d'entiers
- Toutes les variations de nombre à virgule flottante (Real = nombres à virgules flottante IEEE)
- Toutes les variations de types de données TIME

Si le résultat de la comparaison est « VRAI », alors la sortie de l'opération vaut « 1 », autrement elle vaut « 0 ». L'entrée IN1 est comparée avec IN2 selon le type de comparaison sélectionné :

- **==** IN1 est égale à IN2
- **<>** IN1 est différent de IN2
- **>** IN1 est supérieur à IN2
- **<** IN1 est inférieur à IN2
- **>=** IN1 est supérieur ou égale à IN2
- **<=** IN1 est inférieur ou égale à IN2.

10.2.3. Copie valeurs

Paramètre	Déclaration	Type de données		Zone de mémoire	Description
		S7-1200	S7-1500		
EN	Input	BOOL	BOOL	I, Q, M, D, L	Entrée de validation
ENO	Output	BOOL	BOOL	I, Q, M, D, L	Sortie de validation
IN	Input	Séquences de bits, entiers, nombres à virgule flottante, DATE, TIME, TOD, DTL, CHAR, STRUCT, ARRAY, types de données CEI, type de données API (UDT)	Séquences de bits, entiers, nombres à virgule flottante, DATE, DT, LDT, S5TIME, TIME, LTIME, TOD, LTOD, DTL, CHAR, STRUCT, ARRAY, TIMER, COUNTER, types de données CEI, type de données API (UDT)	I, Q, M, D, L ou constante	Valeur source
OUT1	Output	Séquences de bits, entiers, nombres à virgule flottante, DATE, TIME, TOD, DTL, CHAR, STRUCT, ARRAY, types de données CEI, type de données API (UDT)	Séquences de bits, entiers, nombres à virgule flottante, DATE, DT, LDT, S5TIME, TIME, LTIME, TOD, LTOD, DTL, CHAR, STRUCT, ARRAY, TIMER, COUNTER, types de données CEI, type de données API (UDT)	I, Q, M, D, L	Opérande dans lequel la valeur source est transférée.

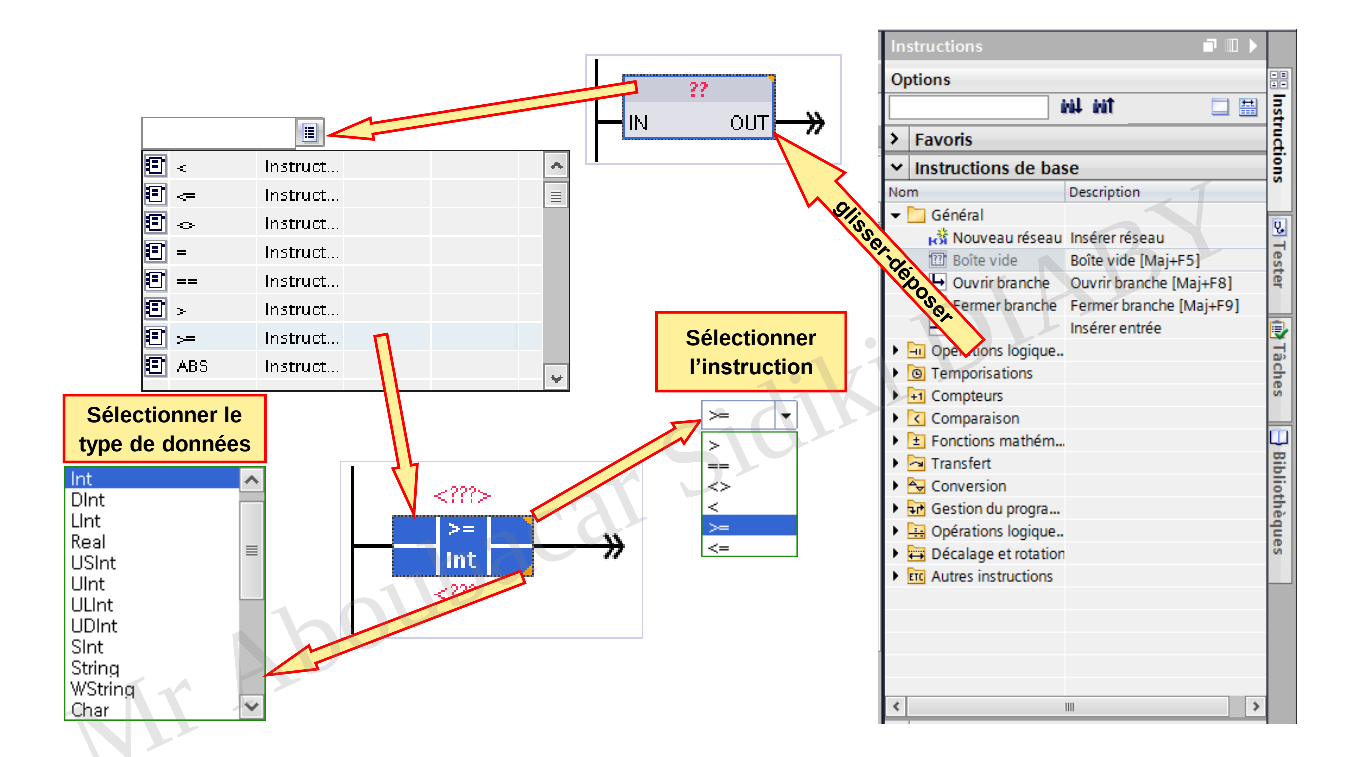


MOVE

Si vous utilisez l’instruction « MOVE » pour transférer le contenu de la variable d’entrée (IN) à la variable de sortie (OUT1).

L’opération n’est exécutée que si l’état du signal sur l’entrée de validation EN est « 1 » ou n’est pas affecté. Dans ce cas et avec un transfert sans erreur, la sortie ENO a également l’état de signal « 1 ».

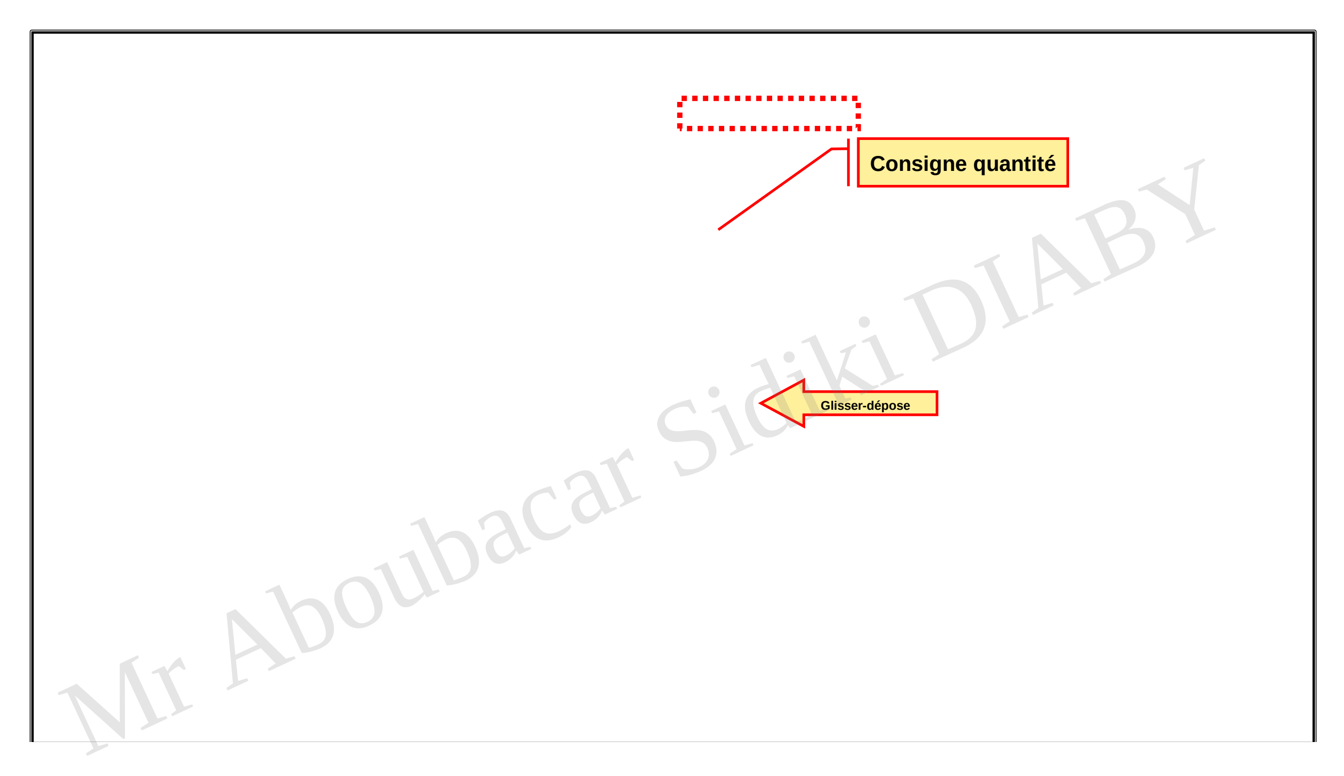
10.2.4. Programmer une instruction via une boîte vide



Programmer une instruction en utilisant « Boite Vide »

Une instruction peut également être programmée dans la "boîte vide". La boîte vide est d'abord tirée de la carte de tâches "Instructions" en utilisant le glisser-déposer, ou elle est tirée des "Favoris" sur le réseau. Ensuite, dans la zone vide, vous sélectionnez l'instruction à utiliser avec quel type de données.

10.2.5. Exercise 1: Comptage des pièces transportées avec le bloc « FB_ComptageADD »



Enoncé

- Lorsque « Q_En_Service » (Q0.1) est activé, les pièces transportées doivent être comptées dès qu'elle franchissement la barrière lumineuse (I3.0) (« B_LS » 0 ->1). Le nombre de pièce transportés (quantité actuel) doit être sauvegardé dans une variable statique.
- Lorsque la valeur de consigner est atteinte, aucune nouvelle séquence de transport ne peut apparaître, ce qui est également visible sur les voyants des postes 1 et 2
- Le compteur peut être acquitté à tout moment via le bouton-poussoir « BP_PosteBL » (I3.4). Lorsque l'opération est activée, c'est-à-dire avec un front positif à « Q_En_Service » (Q0.1), la grandeur réelle est également remise à 0.

Marche à suivre :

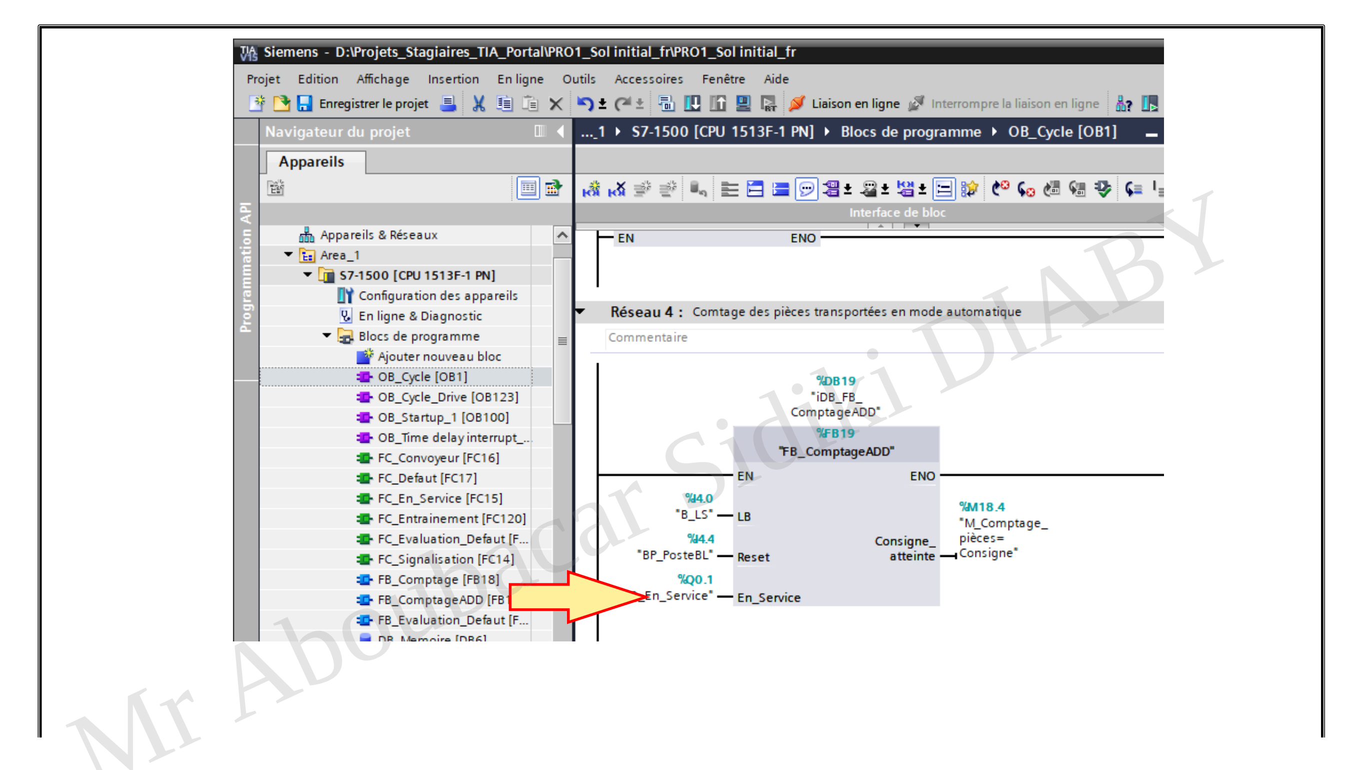
1. Créer un nouveau bloc de fonction « FB_ComptageAdd »

2. Déclarer les paramètres et variables nécessaire

Entrée:	- LB	BOOL,	Signal de la barrière lumineuse
	- Reset	BOOL,	Réinitialiser le nombre des pièces à 0.
	- En_Service	BOOL,	Mode d'opération
Sortie:	- Consigne_atteinte	BOOL,	le nombre de pièce déjà transporté
Statique:	- M_Aux_Front_LB	BOOL,	bit auxiliaire pour l'évaluation du bord du signal de la barrière lumineuse
	- Nb_pieces	UINT,	nombre actuel de pièces transportées
Constante:	- consigne	UINT,	consigne du nombre de pièce à transporter

3. Programmez le nouveau bloc "FB_ComtageAdd" avec les opérations arithmétiques appropriées et n'utilisez aucune variable globale pour le faire. Pour l'évaluation du front de la barrière lumineuse, utilisez la variable statique #Front_Pos et enregistrez la quantité actuelle des pièces transportées dans la variable statique #Nb_pieces.

10.2.6. Exercice 2: Appel du bloc "FB_ComptageAdd



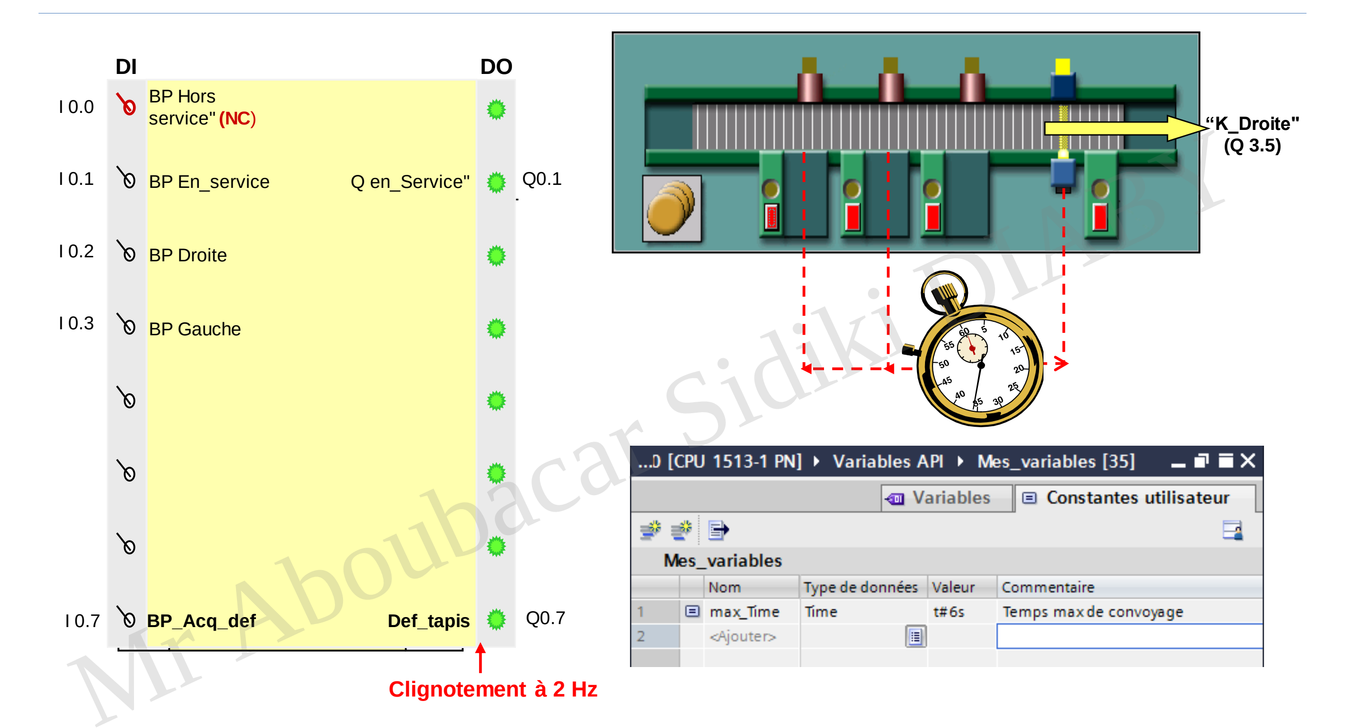
Enoncé

Vous devez appeler le bloc « FB_ComptageAdd » et lui assigner les paramètres adéquats.

Marche à suivre :

1. Dans OB_Cycle, appeler le nouveau bloc « FB_ComptageAdd »
2. Fournir les parameters formels avec les paramètres réels pertinents (voir image ci-dessus)
3. Modifier les blocs « FC_Convoyeur » et « FC_Signalisation » de telle sorte que lorsque la consigne est atteinte, il n'est plus possible de transporter une pièce, les voyants des postes 1 et 2 sont éteints et le voyant de la barrière lumineuse « Q_PosteLS » (Q3.4) clignote à une fréquence de 1Hz. Pour cela, utilisez la variable globale « M_Comptage_pièces=Consigne».
4. Compiler et sauvegarder votre projet et vérifier qu'il fonctionne correctement.

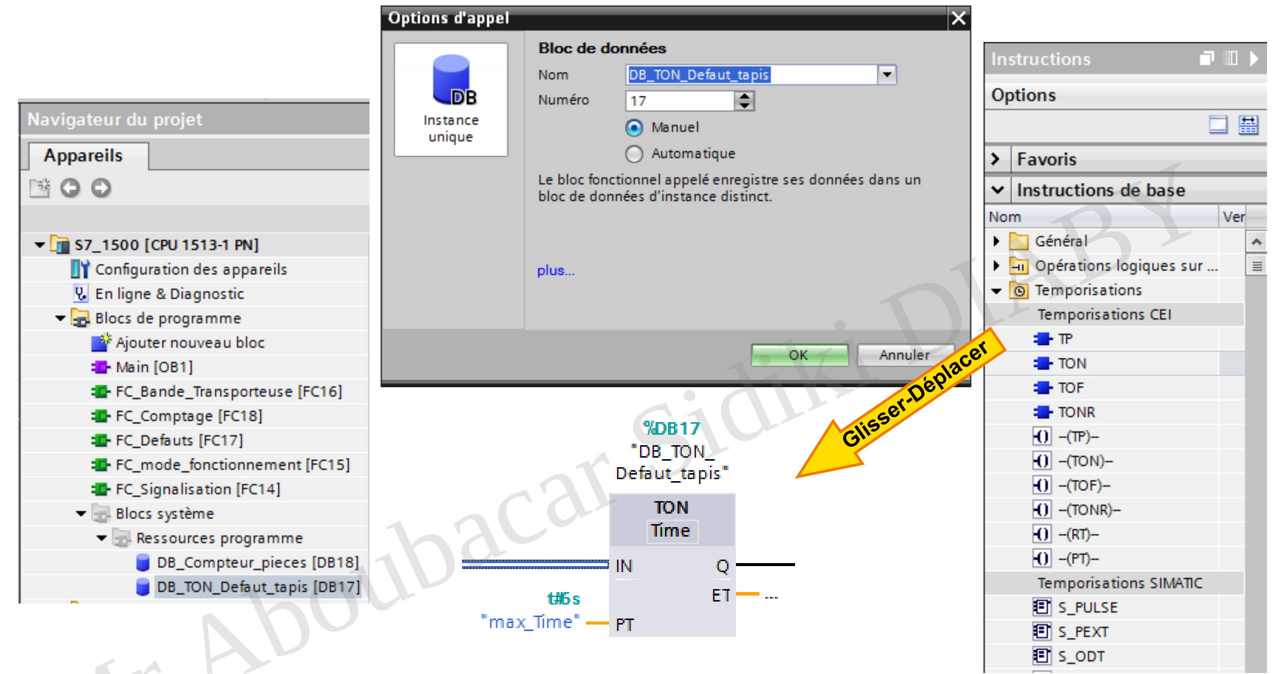
10.3. Enoncé : Surveillance temporisée des séquences de transport et comptage des pièces à l'aide des fonctions CEI



Enoncé

1. Les séquences de transport automatique doivent être surveillées dans le temps à l'aide d'une fonction CEI. La surveillance doit fonctionner comme suit:
   - Si une séquence de transport dure plus longtemps que le temps de surveillance de 6 secondes, il y a un défaut et le moteur du convoyeur est automatiquement désactivé.
   - Un défaut est affiché avec un clignotement de 2Hz sur la LED du simulateur "Def-tapis" (Q0.7).
   - Un défaut peut être acquitté via le commutateur du simulateur "BP_Acq_Def" (I 0.7).
   - Tant qu'il y a un défaut non reconnu, les voyants "P_poste1" (Q3.1) et "P_poste2" (Q3.2) sont sombres et aucune nouvelle séquence de transport ne peut être démarrée.
2. Le comptage des pièces transportées doit être réalisé avec une fonction CEI.

10.3.1. Temporisation CEI



Remarque : la durée de temporisation doit être positive

Bloc de donnée

En plus des variables requises en interne, la fonction de temporisation stocke également l'heure actuelle déjà expirée dans un bloc de données qui doit être spécifié lors de la programmation de la fonction de temporisation. Le bloc de données spécifié est automatiquement généré par l'éditeur avec exactement la structure interne requise par la fonction de temporisation. L'utilisateur n'a pas d'autre effort de programmation avec ce bloc de données que de devoir le télécharger dans la CPU.

Variable d'entrée "PT"

La variable PT de la fonction temps peut-être du type Time ou LTime.

Type de donnée TIME

Le contenu d'une variable ou d'une constante du type de données TIME est interprété comme un nombre entier en millisecondes et stocké dans la mémoire sous la forme d'un entier de 32 bits avec signe. La représentation contient des informations sur les jours (d), les heures (h), les minutes (m), les secondes (s) et les millisecondes (ms).

Plage de valeurs : T#-24d20h31m23s648ms à T#+24d20h31m23s647ms

Type de donnée LTIME

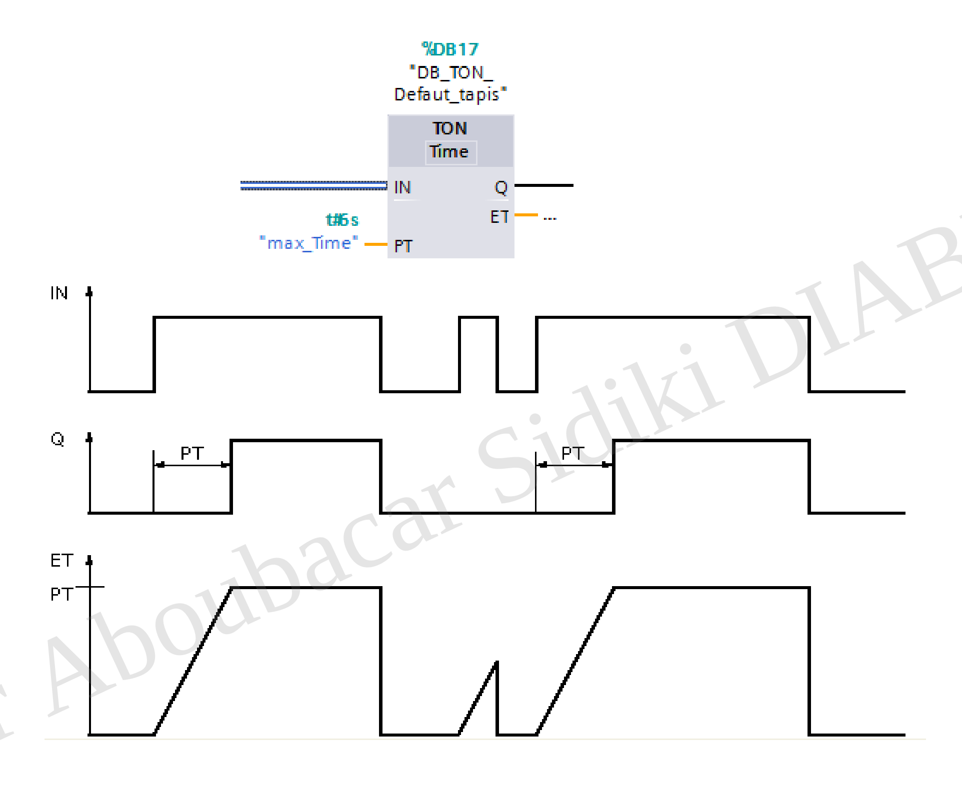
Le contenu d'une variable du type de données LTIME est interprété comme des nanosecondes. La représentation contient des informations sur les jours (d), les heures (h), les minutes (m), les secondes (s), les millisecondes (ms), les microsecondes (us) et les nanosecondes (ns).

Plage de valeur: -106751d 23h 47m 16s 854ms 775us 808ns

à

+106751d 23h 47m 16s 854ms 775us 807ns

10.3.2. Temporisations TON : Chronogramme

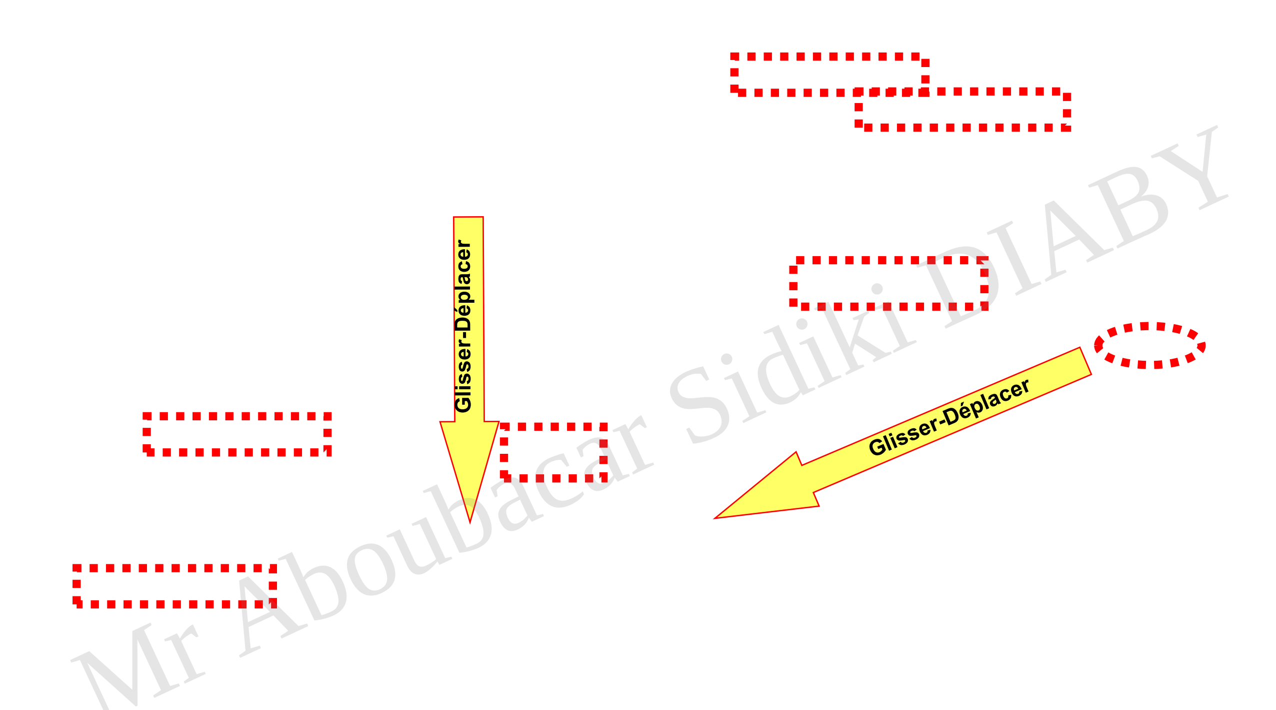


TON

La fonction de temporisation TON (Temporisation de temporisation, "ON Delay") est démarrée avec un front montant sur l'entrée IN. Pour que le temps expire, RLO doit continuer à être '1'. La fonction de minuterie fournit un signal '1' à la sortie Q, dès que l'heure spécifiée (variable ou constante du type de données TIME ou LTIME) à l'entrée PT a expiré et tant que le signal de départ à l'entrée IN existe encore. Le temps déjà expiré peut être interrogé à la sortie ET en passant une variable de type de données TIME ou LTIME.

Parameter	Data type	Memory area	Description
IN	BOOL	I, Q, M, D, L	Entrée de démarrage
PT	TIME	I, Q, M, D, L or constant	Durée d'un retard au front montant de IN PT>0
Q	BOOL	I, Q, M, D, L	Sortie mise à 1 après écoulement d'une durée PT
ET	TIME	I, Q, M, D, L	Valeur du temps actuel

10.3.3. Exercise 3: Surveillance du temps de transport avec le bloc « FC_Defaut »



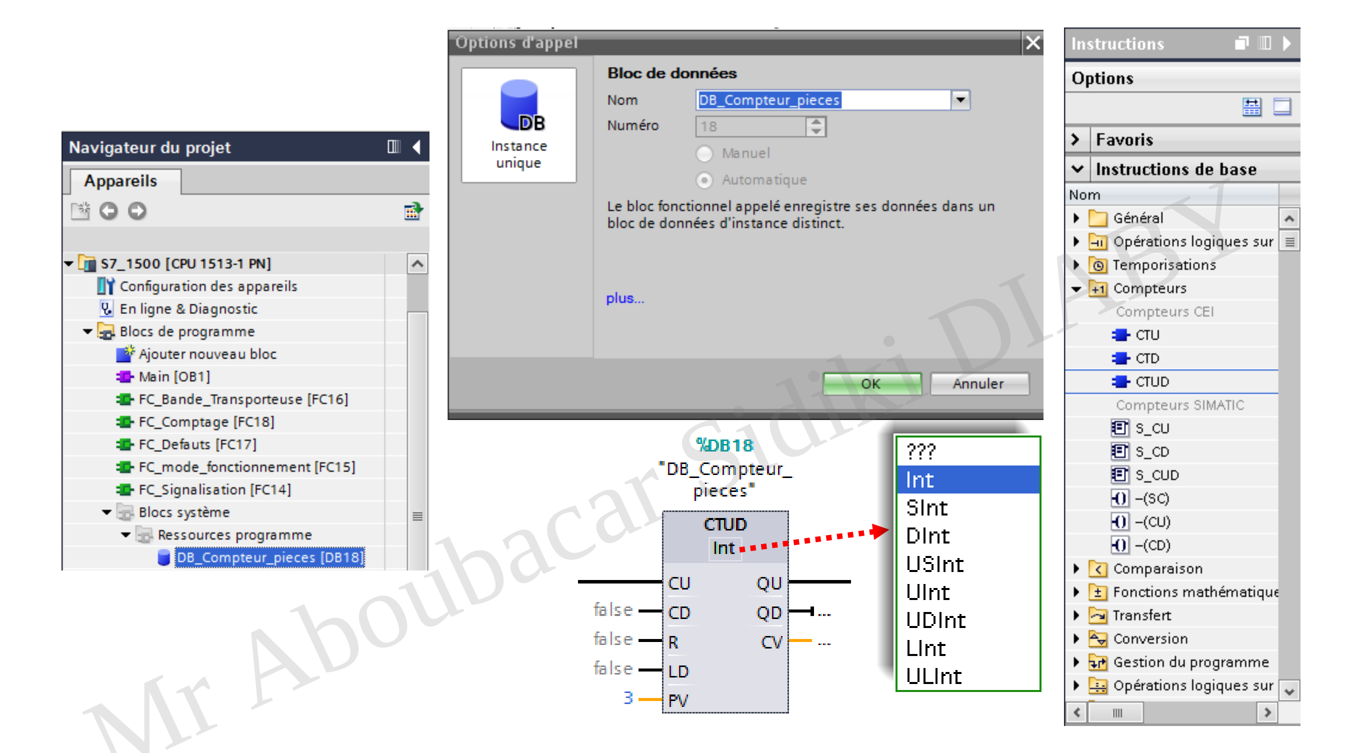
Enoncé

Le temps des opérations de transport doit être surveillé comme décrit plus haut. Lorsqu'une opération de transport dépasse le temps maximal autorisé de 6 secondes, le moteur de la bande est automatiquement coupé et le défaut est signalé par un clignotement à 2 Hz de « Def_tapis» (Q0.7) sur le simulateur. Une nouvelle opération de transport ne peut être lancée qu'après acquittement du défaut via « BP_Acq_defaut» du simulateur (I0.7).

Marche à suivre

1. Créez la constante utilisateur « *max_Time* » du type de données *Time* avec la valeur *T#6s*, comme indiqué sur la figure.
2. Développez le bloc "FC_Defaut" avec les fonctions nécessaires :
   – A l'entrée IN, programmez les conditions de démarrage appropriées (la minuterie doit être démarrée lorsqu'une séquence de transport automatique commence)
   – Transmettre le bloc de données "iDB_IEC_Timer_1" en tant que DB d'instance à la fonction de temporisation CEI TON et, en tant que durée, la constante utilisateur "maxTime" (voir image)
   – Dans le cas où le temps de transport maximum est dépassé, paramétrez le bit de mémoire "M_Defaut_Tapist" (M17.0), afin de pouvoir le relier plus logiquement dans d'autres blocs plus tard.
3. Programmez l'appel du « FC_Defaut » (FC17) dans le « Main » (OB1).
4. Dans le bloc « FC_Convoyeur» (FC16), programmez la coupure sur défaut du moteur de la bande en reliant en conséquence le mémento « M_Defaut_tapis» (M17.0).
5. Programmez les verrouillages correspondants dans « FC_Signalisation » (FC14) afin que les LED de signalisation «Q_Poste1» (Q3.1) et « Q_Poste2» (Q3.2) soient éteintes en présence d'un défaut « M_Defaut_tapis» (M17.0).
6. Chargez les blocs modifiés dans la CPU et vérifiez le fonctionnement correct du programme.
7. Enregistrez votre projet.

10.4. Compteurs CEI : CTU, CTD, CTUD



Compteurs

Les compteurs sont utilisés pour compter les événements, enregistrer les quantités, etc. Il y a des compteurs et des compteurs descendants ainsi que des compteurs qui peuvent compter dans les deux directions.

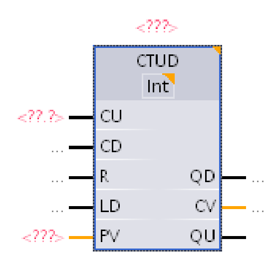
Plage de valeur du compteur

Le nombre ou la plage de valeurs d'un compteur dépend de son type de données (voir image) qui est toujours un nombre entier. Les différents types de données Integer sélectionnables se différencient simplement dans leur plage de valeurs et déterminent ainsi la plage de comptage du compteur.

Bloc de données d’instance

En plus des variables requises en interne, le compteur stocke également sa valeur de compteur actuelle dans un bloc de données appelé instance, qui doit être spécifié lors de la programmation d'un compteur. Le bloc de données d'instance spécifié est généré automatiquement par l'éditeur avec exactement la structure interne requise par le compteur. L'utilisateur n'a pas d'autre effort de programmation avec ce bloc de données que de devoir le télécharger dans la CPU.

10.4.1. Compteurs CEI HAUT/BAS : Entrées



Paramètre	Déclaration	Type de données	Zone de mémoire	Description
CU	Input	BOOL	I, Q, M, D, L ou constante	Entrée de comptage
CD	Input	BOOL	I, Q, M, D, L ou constante	Entrée de décomptage
R	Input	BOOL	I, Q, M, D, L, T, C, P ou constante	Entrée de remise à zéro
LD	Input	BOOL	I, Q, M, D, L, T, C, P ou constante	Entrée de chargement
PV	Input	Nombres entiers	I, Q, M, D, L, P ou constante	Valeur pour laquelle la sortie QU est mise à 1.

Entrée CU et CD

Avec un front positif à l'entrée CU, le compte actuel est augmenté de un; avec un front positif sur le CD d'entrée, le compte courant est diminué de 1. Cela signifie que l'utilisateur n'a pas besoin de programmer une évaluation de front.

Si un front positif est détecté sur les deux entrées simultanément ou dans le même cycle, le compte courant reste inchangé. Si la limite supérieure ou inférieure du type de données spécifié est atteinte, le nombre n'est plus augmenté ni diminué pour un front positif à CU ou CD.

Entrée R

The input R acts statically, that is, as long as RLO '1' is at input R, the count is set to 0 and rising edges or RLO '1' at the inputs CU, CD and LOAD have no effect on the current count.

L'entrée R agit de manière statique, c'est-à-dire tant que RLO '1' est à l'entrée R, le compte est mis à 0 et les fronts montants ou RLO '1' aux entrées CU, CD et LOAD n'ont aucun effet sur le compte actuel.

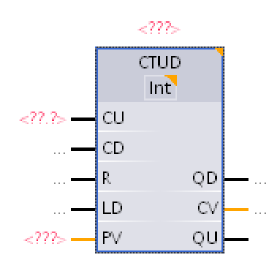
Entrée LD (Charger, uniquement pour les compteurs descendants)

L'entrée LD agit statiquement, c'est-à-dire que tant que RLO '1' est à l'entrée LOAD, le compte actuel est défini sur la valeur passée à l'entrée PV et les fronts positifs aux entrées CU et CD n'ont aucun effet sur le décompte.

Entrée PV

La valeur à laquelle le compte doit être défini doit être transmise à l'entrée PV tant que RLO '1' est à l'entrée LD. La variable ou la constante transmise à l'entrée doit être compatible avec le type de données du compteur.

10.4.2. Compteurs CEI HAUT/BAS : Sorties



Paramètre	Déclaration	Type de données	Zone de mémoire	Description
QU	Output	BOOL	I, Q, M, D, L	Etat du compteur
QD	Output	BOOL	I, Q, M, D, L	Etat du décompteur
CV	Output	Nombres entiers, CHAR, DATE	I, Q, M, D, L, P	Valeur de comptage actuelle

Sortie QU

L'état actuel du compteur peut être vérifié à la sortie QU. Tant que le nombre actuel est supérieur ou égal à la valeur du paramètre PV, la sortie QU a le statut '1', sinon, le statut '0'.

Sortie QD (Seulement pour les décompteurs)

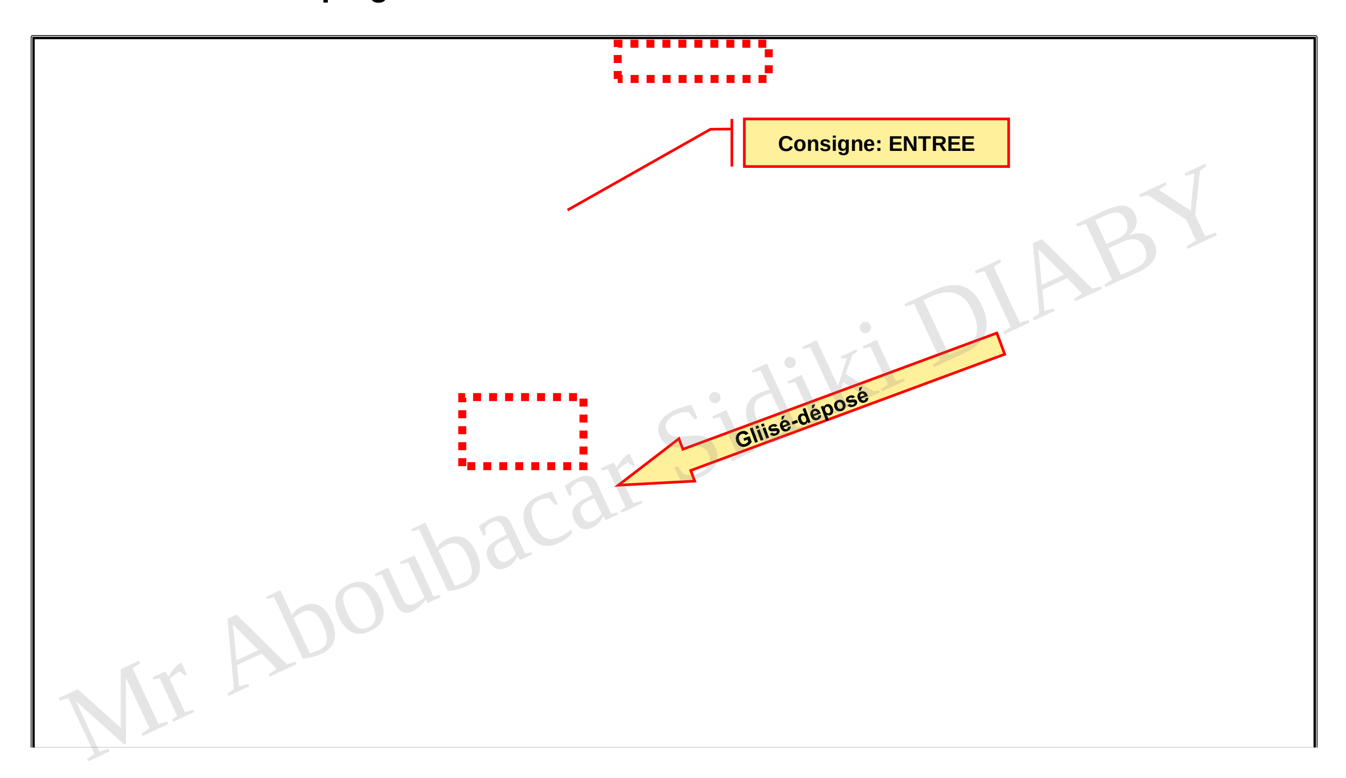
L'état actuel du décompteur peut être vérifié à la sortie QD. Tant que le nombre actuel est inférieur ou égal à zéro, la sortie QD a le statut "1", sinon le statut "0".

Sortie CV

Le nombre actuel est sorti à la sortie CV. La variable transmise à la sortie doit être compatible avec le type de données du compteur.

Mr Aboubacar Sidiki DIABY

10.4.3. Exercice 4: Comptage des pièces transportées avec le bloc « FB_Comptage »



Enoncé

- Lorsque "Q_En_Service" (Q0.1) est activé, les pièces transportées doivent être comptées dès qu'elles ont franchi la barrière lumineuse "B_LB" (I 3.0) ("B_LB" 0-> 1). Le nombre de pièces transportées doit être enregistré avec un compteur et stocké dans la variable statique #ACT.
- Lorsque la grandeur de consigne (valeur constante 3) est atteinte, aucune nouvelle séquence de transport ne peut apparaître, ce qui est également visible sur les LED "P_Poste1" et "P_Poste2
- Le compteur peut être acquitté (réinitialisé à 0) à tout moment via le bouton-poussoir "BP_Poste_BL" (I 3.4). Lorsque l'opération est activée, c'est-à-dire avec un front positif à "Q_En_Service" (Q0.1), la nb de pièces est remis à 0.

Marche à suivre :

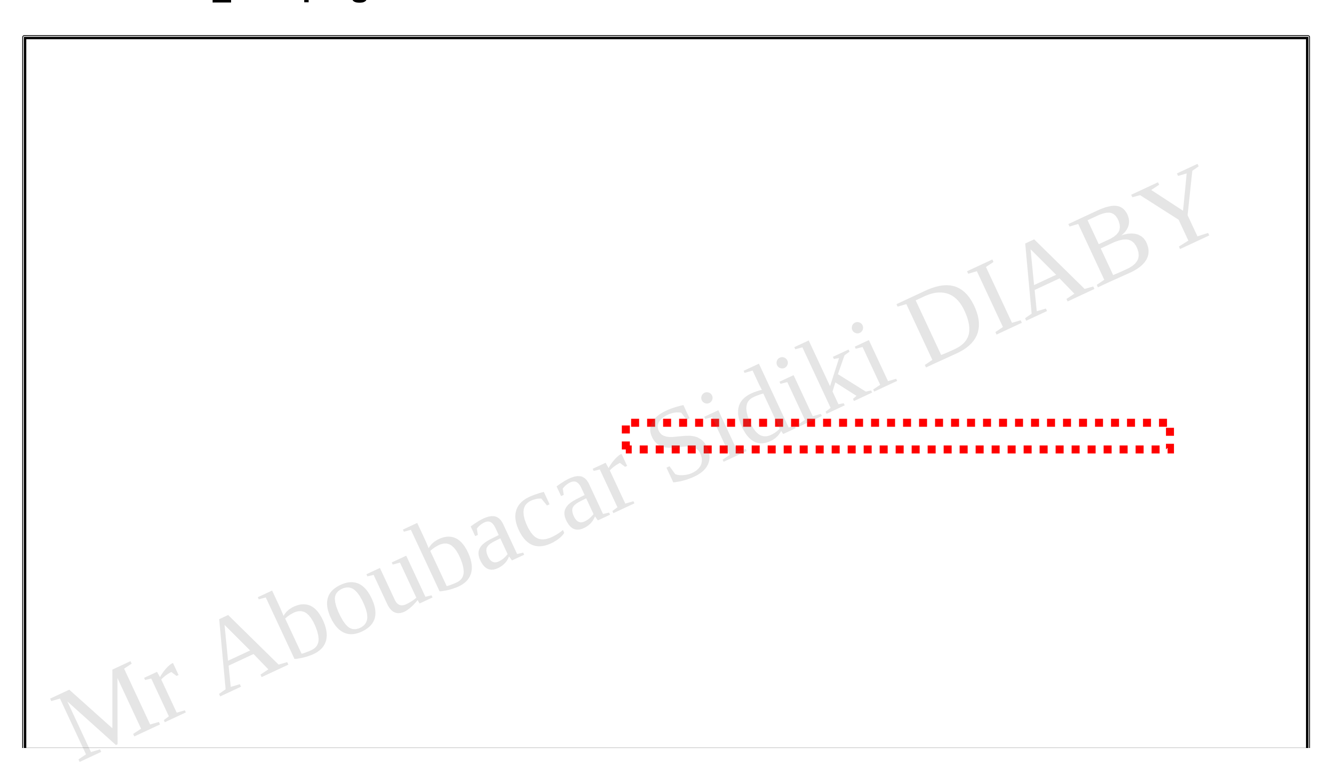
4. Créez le nouveau bloc « FB_Comptage »

5. Déclarez les paramètres et variables nécessaires:

Entrée:	- LB	BOOL,	Signal de la barrière lumineuse
	- Reset	BOOL,	Réinitialiser le nombre des pièces à 0.
	- En_Service	BOOL,	Mode d'opération
	- Consigne	UINT,	consigne du nombre de pièce à transporter
Sortie:	- Consigne_atteinte	BOOL,	le nombre de pièce déjà transporté
Statique:	- M_Aux_Front_LB	BOOL,	bit auxiliaire pour l'évaluation du front du signal de la barrière lumineuse
	- M_Aux_Front_En_Service	BOOL	bit auxiliaire pour l'évaluation du front du signal de le mise en Service
	- Nb_pieces	UINT,	nombre actuel de pièces transportées -

6. Programmez le nouveau bloc "FB_Comptage" avec les opérations arithmétiques appropriées et n'utilisez pas de variables globales (tags) pour le faire. Pour l'évaluation du bord de la barrière lumineuse, utilisez la variable statique #M_Aux_Front_LB et enregistrez la quantité actuelle des pièces transportées dans la variable statique #Nb_pieces.

10.4.4. Exercice 5: Remplacement de l'appel "FB_ComptageADD" par l'appel de "FB_Comptage"



Marche à suivre :

1. Dans la table de variables "Mes_Variables", créez une nouvelle balise "Const_Consigne" du type UINT et attribuez-lui la valeur "3".

2. Remplacez l'appel de "FB_ComptageADD" par l'appel de "FB_Comptage". Utilisez pour cela la fonction "Changer l'appel de bloc" dans laquelle les paramètres réels sont adoptés.

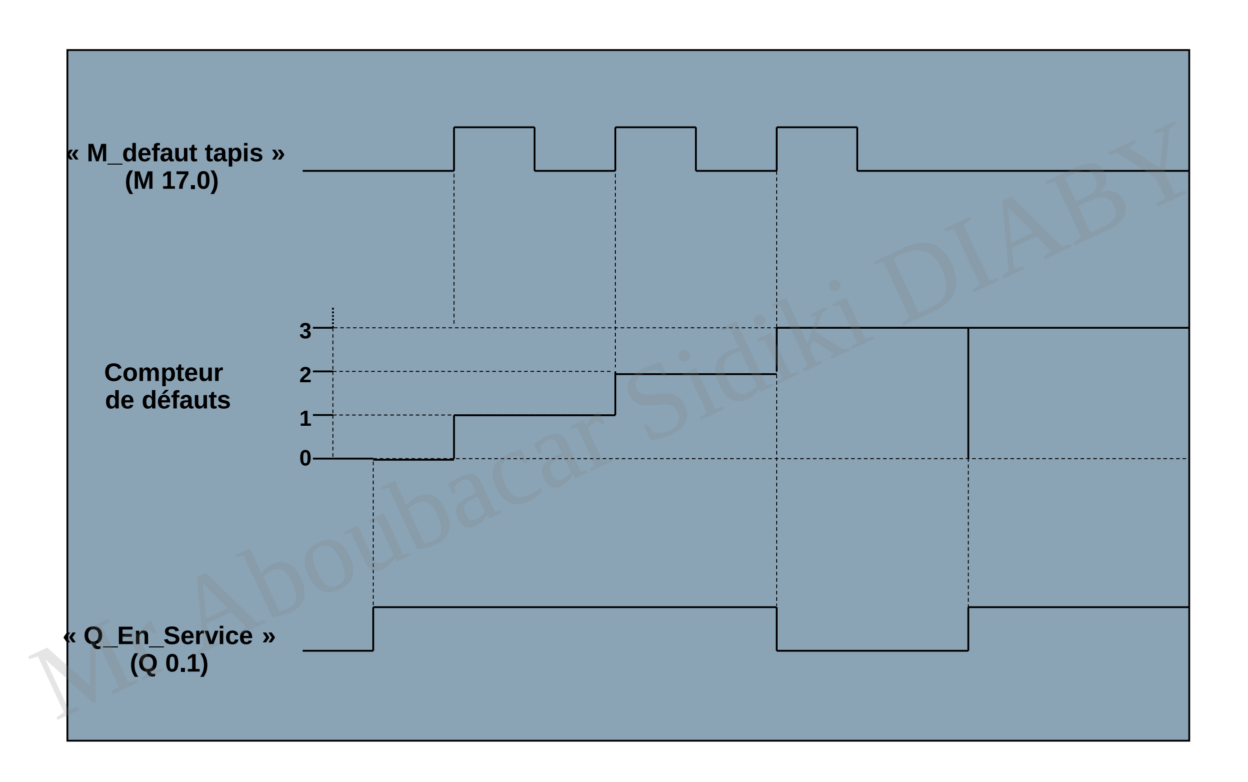
3. Créez ne nouvelle instance (iDB_FB_ComptageADD) pour les appels de bloc en cliquant avec le bouton droit de la souris sur l'appel du FB, puis en sélectionnant l'élément de menu contextuel "Changer l'instance ...".

4. Interconnectez le nouveau paramètre formel d'entrée "Consigne" avec la nouvelle constante globale "const_consigne"

5. Charger les blocs modifiés dans la CPU et vérifiez la fonction du programme.

6. Sauvegardez votre projet.

10.5. Exercice complémentaire 6: Comptage des défauts de convoyeur en développant "FC_Defaut"



Fonction existante

Le temps des opérations de transport est surveillé lorsque « Q_En_Service » (Q0.1) est activé. Lorsqu'une opération de transport dépasse le temps maximal autorisé de 6 secondes, un signal de défaut bande est émis et le moteur de la bande est automatiquement coupé (liaison logique dans « FC_Convoyeur » (FC16)).

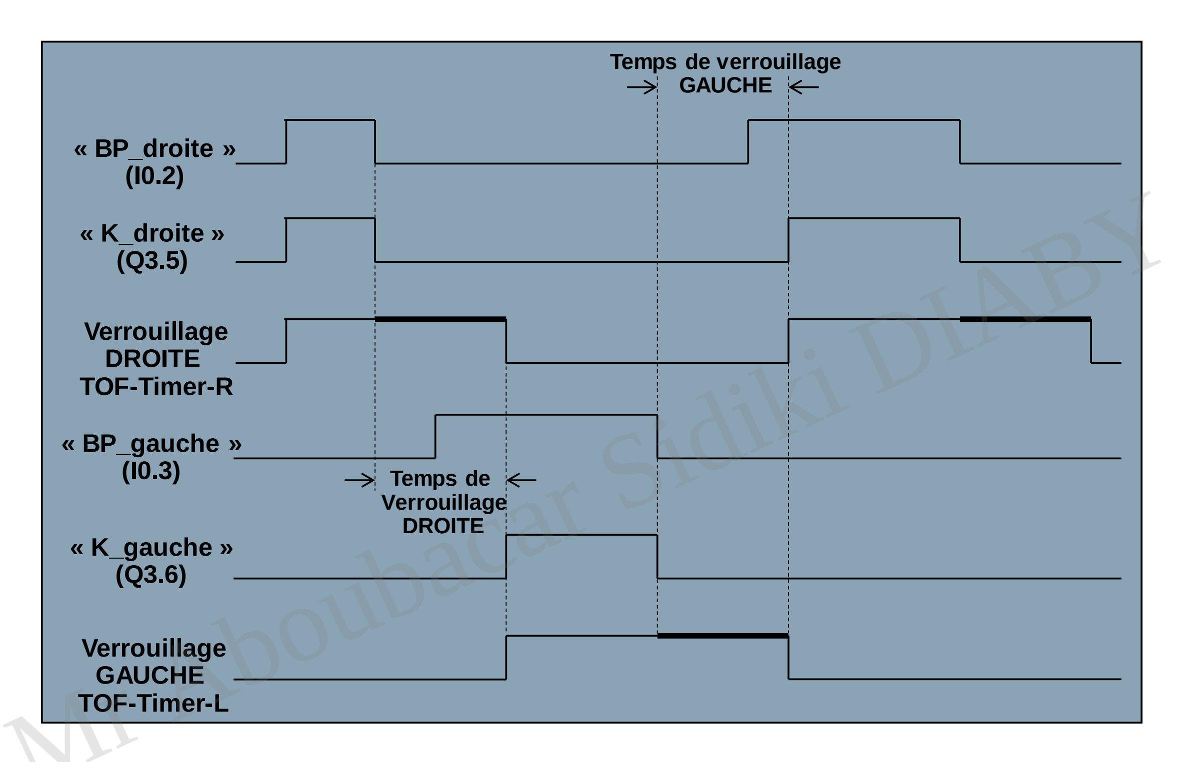
Enoncé :

Lorsque " Q_En_Service " (Q0.1) est activé, les défauts du convoyeur doivent être comptés. Après 3 défauts de convoyeur, "Q_En_Service" (Q0.1) doit être désactivé pour des raisons de sécurité. Pour démarrer une nouvelle séquence de transport, le défaut (comme déjà programmé) doit être acquitté et " Q_En_Service " (Q0.1) doit être réactivé.

Marche à suivre :

1. Dans "FC_Defaut dans un nouveau réseau, programmer le comptage des défauts du convoyeur Le compteur compte" 1 "chaque fois qu'un défaut de convoyeur se produit ("M_Defaut_tapis "(M17.0) =" 1 ").
2. Dans "FC_ Defaut ", passez la valeur à la sortie du compteur Q à n'importe quel bit de mémoire que vous aimez.
3. Dans "FC_En_Service", programmez l'arrêt (reset) de " Q_En_Service " (Q0.1) après trois défauts de convoyeur. Pour cela, utilisez le bit de mémoire auquel vous avez passé la sortie du compteur Q dans "FC_ Defaut ".
4. Charger les blocs modifiés dans la CPU et vérifiez la fonction du programme.
5. Sauvegardez votre projet.

10.5.1.1. Exercice complémentaire 7 : Temps de verrouillage pour la marche par impulsions



Fonction existante

Lorsque « Q_En_Service » (Q0.1) est désactivé, le moteur de la bande peut être commandé par impulsions vers la droite ou la gauche à l'aide des boutons-poussoirs « BP_droite » (I0.2) et « BP_gauche » (I0.3) du simulateur.

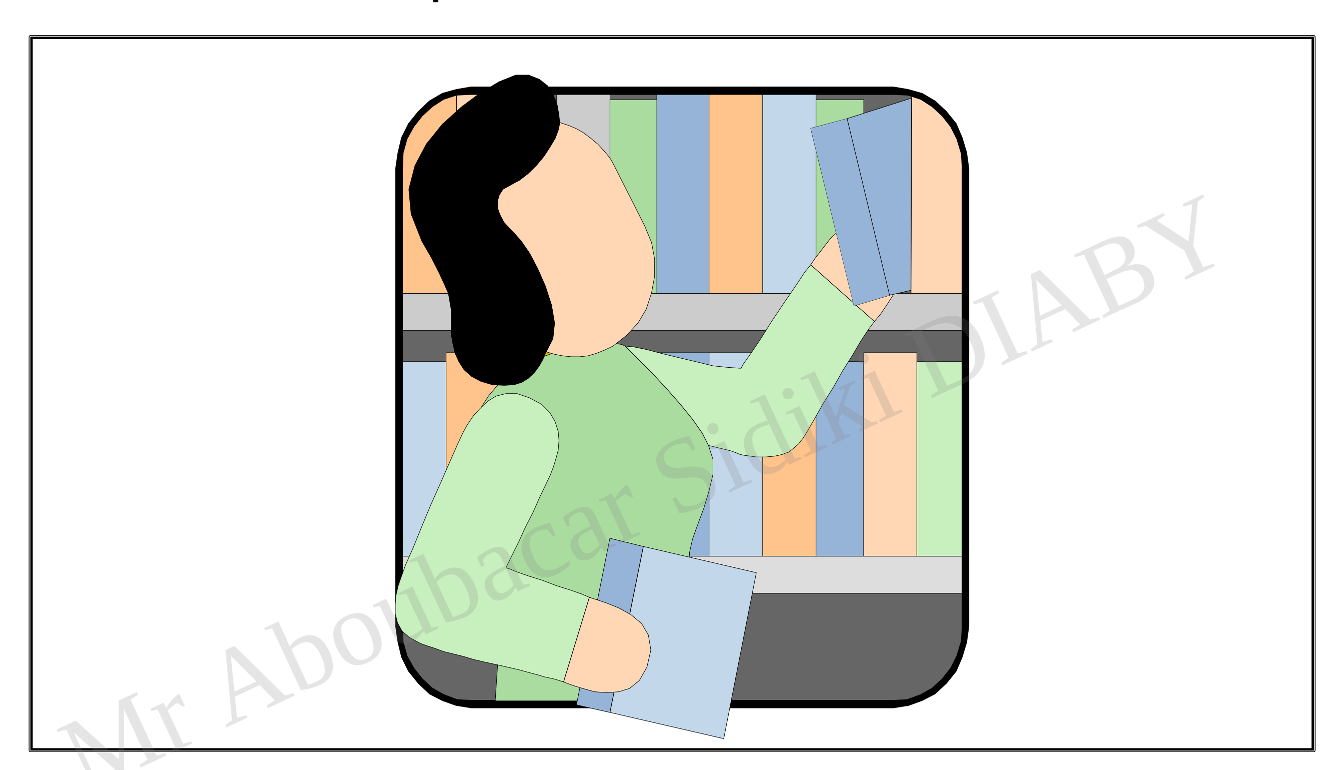
Enoncé

Pour éviter les à-coups de charge trop brutaux, une commande en sens inverse ne doit être possible qu'après un délai de 2 secondes après une commande par impulsions vers la droite ou la gauche (voir figure). Si un déplacement de la bande vers la droite a, par ex., été commandé, un déplacement vers la gauche ne sera donc autorisé qu'après écoulement du temps de verrouillage de 2 secondes.

Marche à suivre

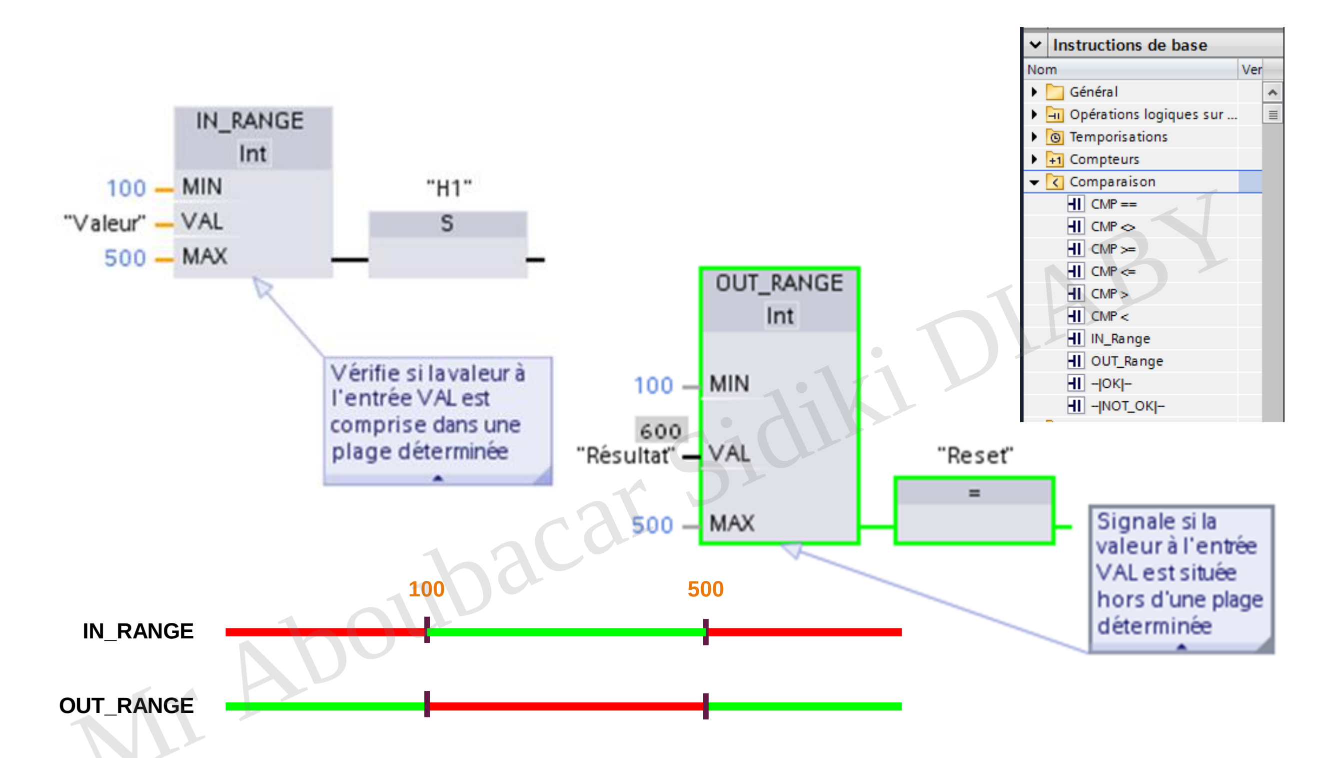
1. Dans « FC_Convoyeur » (FC16), programmez un temporisateur TOF (retard à la retombée) pour le temps de verrouillage vers la droite et un autre pour le temps de verrouillage vers la gauche, et affectez chaque résultat Q à un mémento.
2. Reliez ce mémento dans les conditions de marche par impulsions.
3. Chargez le « FC_Convoyeur » (FC16) modifié dans la CPU et vérifiez le fonctionnement du programme.

10.6. Pour en savoir plus…





10.6.1. Opérations du comparateur : IN_RANGE, OUT_RANGE



IN_RANGE

Vous pouvez utiliser l'instruction "Value within range" pour déterminer si la valeur de l'entrée VAL est comprise dans une plage de valeurs spécifique. Vous spécifiez les limites de la plage de valeurs avec les entrées MIN et MAX (paramètres). Si la valeur à l'entrée VAL remplit la comparaison MIN <= VAL ou VAL <= MAX, la sortie de la boîte a l'état du signal "1". Si la comparaison n'est pas réalisée, la sortie de la boîte a l'état du signal "0".

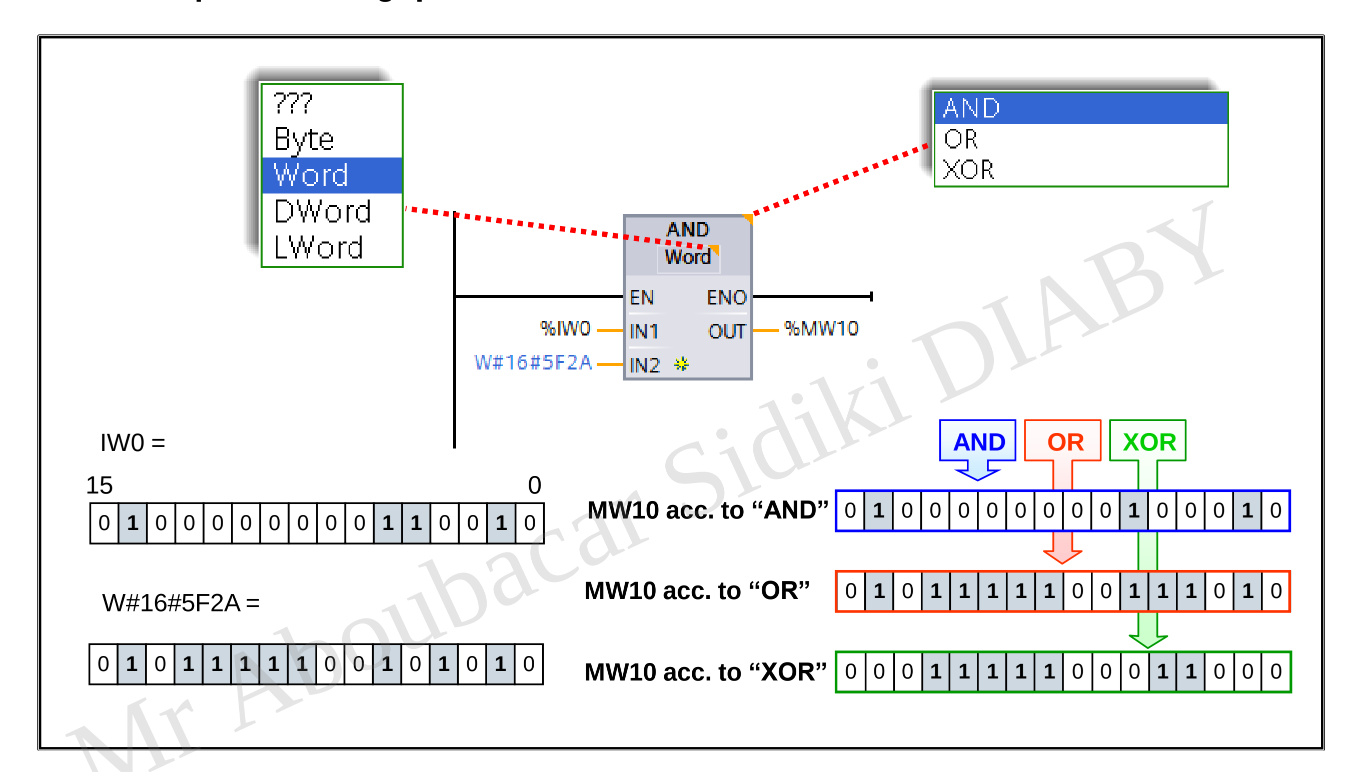
La fonction de comparaison ne peut être exécutée que si les valeurs à comparer sont du même type de données.

OUT_RANGE

Vous pouvez utiliser l'instruction "Value outside range" pour déterminer si la valeur de l'entrée VAL est en dehors d'une plage de valeurs spécifique. Vous spécifiez les limites de la plage de valeurs avec les entrées MIN et MAX (paramètres). Si la valeur à l'entrée VAL remplit la comparaison MIN> VAL ou VAL> MAX, la sortie de la boîte a l'état du signal "1". Si la comparaison n'est pas réalisée, la sortie de la boîte a l'état du signal "0".

La fonction de comparaison ne peut être exécutée que si les valeurs à comparer sont du même type de données.

10.6.2. Opérations logiques combinatoires



AND

L'instruction "AND" associe (combine) les deux valeurs numériques aux entrées IN1 et IN2 bit par bit conformément à la table de vérité ET. Le résultat de l'opération ET est stocké à la sortie OUT. L'instruction est exécutée lorsque EN = 1.

Exemple: Masquer la 4ème décade des boutons de la molette:

IW2= = 0100 0100 1100 0100
W#16#0FFF = 0000 1111 1111 1111

MW30 = 0000 0100 1100 0100

OR

L'instruction "Or" associe (combine) les deux valeurs numériques aux entrées IN1 et IN2, bit par bit, conformément à la table de vérité OR. Le résultat de l'opération OU est stocké à la sortie OUT. L'instruction est exécutée lorsque EN = 1.

Exemple: Définition du bit 0 dans MW32:

MW32 = 0100 0010 0110 1010
W#16#0001 = 0000 0000 0000 0001

MW32 = 0100 0010 0110 1011

XOR

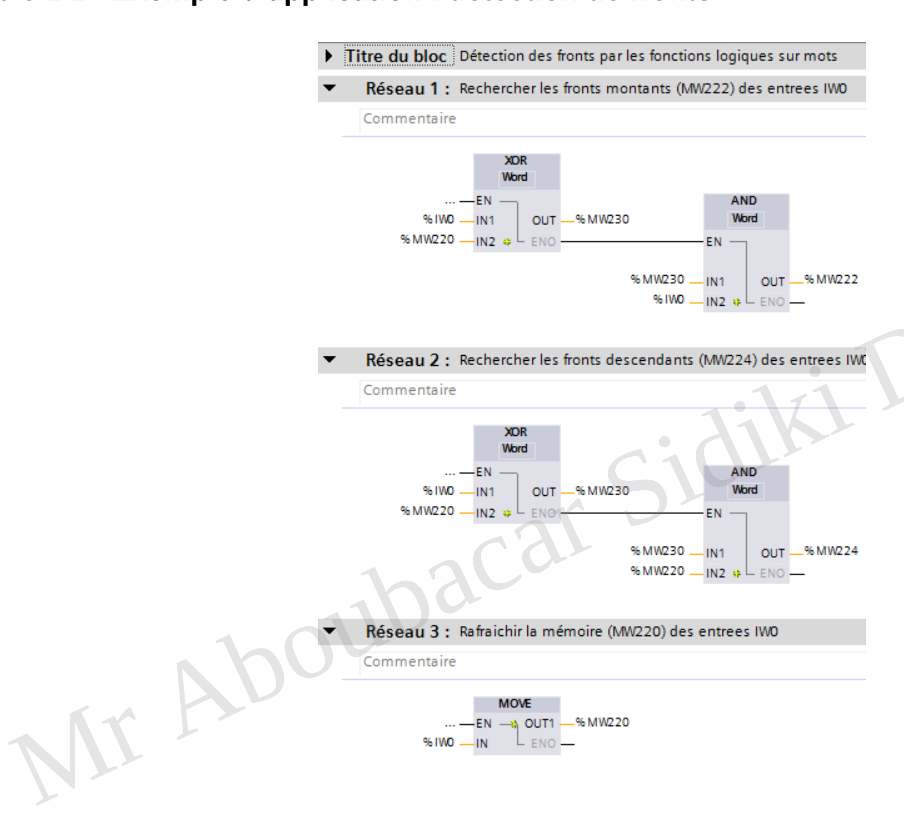
L'instruction "OU exclusif" associe (combine) les deux valeurs numériques aux entrées IN1 et IN2 bit par bit conformément à la table de vérité XOR. Le résultat de l'opération XOR est stocké à la sortie OUT. L'instruction est exécutée lorsque EN = 1.

Exemple: Détection des changements de signal dans IW0:

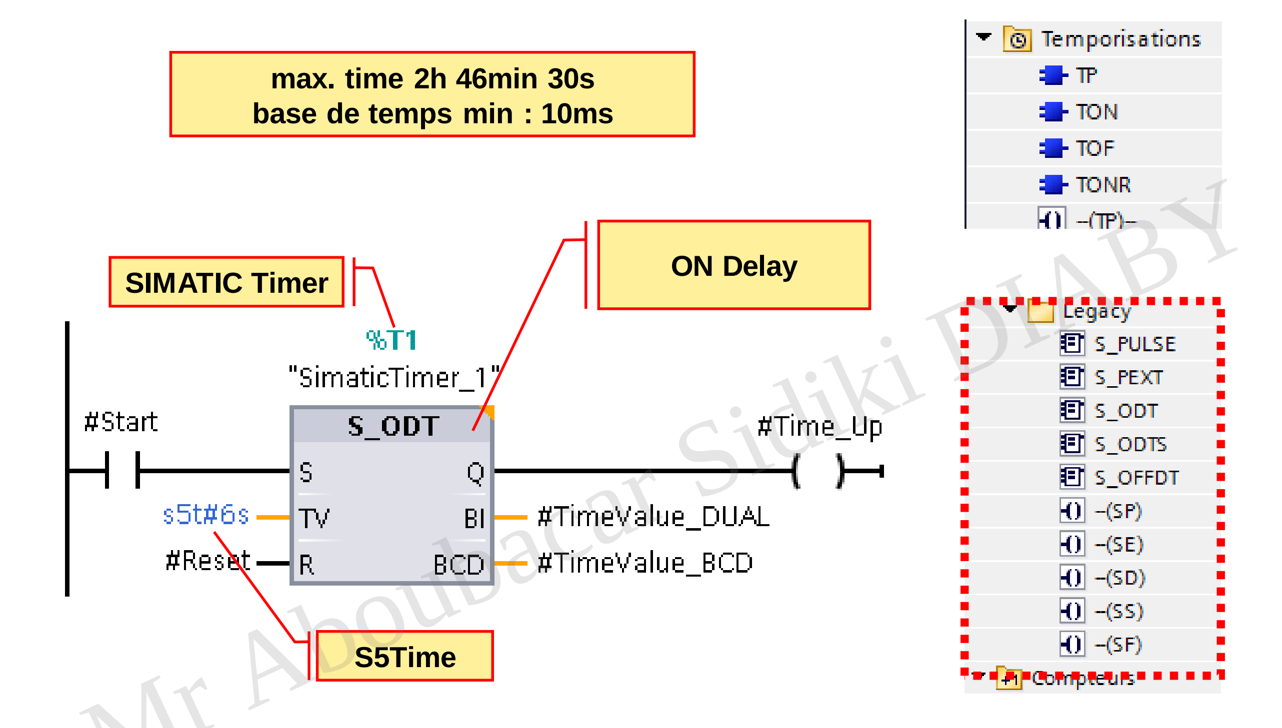
IW0 = 0100 0100 1100 1010
MW28 = 0110 0010 1011 1001

MW24 = 0010 0110 0111 0011

10.6.2.1. Exemple d'application : détection de fronts



10.6.3. SIMATIC-Temporisation



Pour des raisons de compatibilité avec STEP5, vous avez la possibilité d'utiliser SIMATIC-Timers dans un S7-1500. Ces minuteurs sont globaux. Pendant le démarrage, la précision de la plage de valeurs et le type de minuterie (ON Delay, OFF Delay, etc.) sont définis.

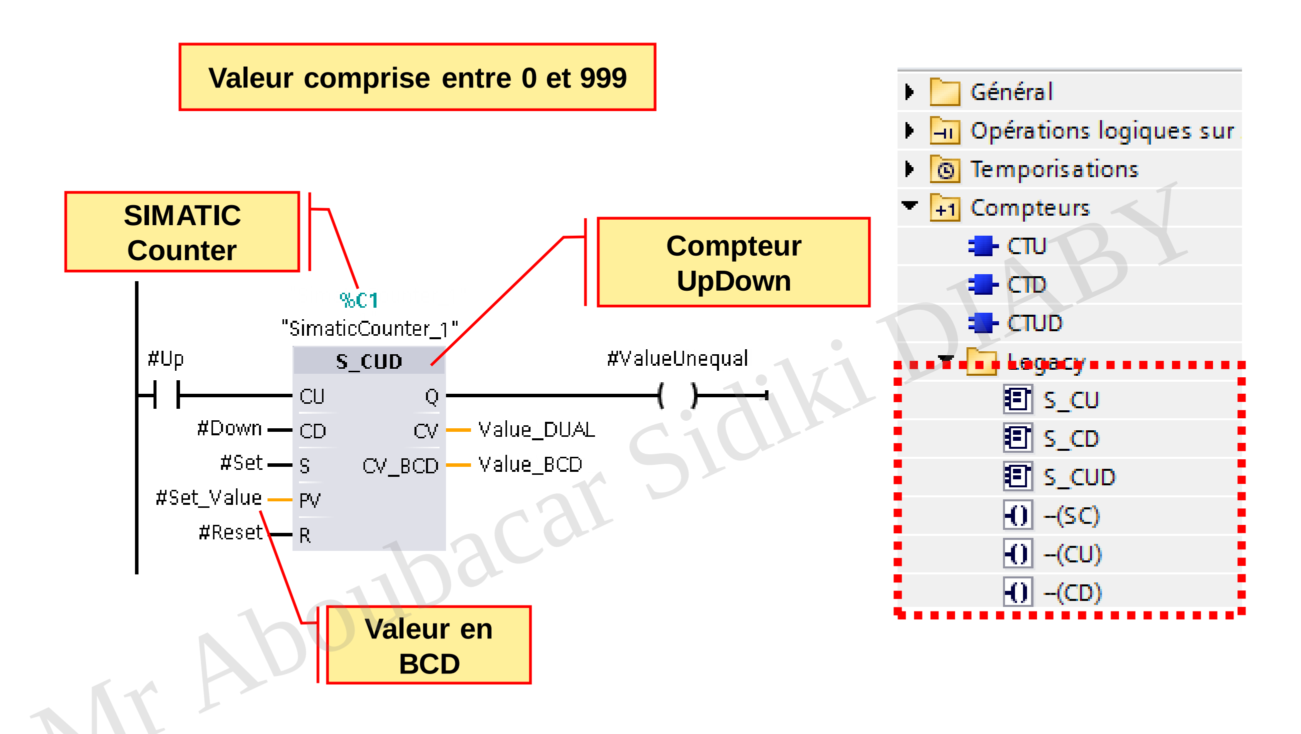
SIMATIC-Temporisations ont les propriétés système suivantes :

- Ils sont affectés à un nombre fixe,
- Ils ne peuvent pas être utilisés à plusieurs reprises,
- Le délai se situe entre 0,01 et 9,99, 0,1 et 99,9, 1 et 999 ou 10 et 9990
- La précision dépend de la durée sélectionnée 10ms, 100ms, 1s ou 10s

Commentaire

Dans la cellule temporelle (dans l'image T1 de la zone de mémoire Timer), le système d'exploitation réduit la valeur temporelle dans un intervalle (intervalle de temps) défini par la base de temps d'une unité respectivement jusqu'à ce que la valeur temporelle soit égale à "0". La réduction se produit de manière asynchrone avec le programme utilisateur. Par conséquent, le temps résultant est au maximum jusqu'à un intervalle de temps de la base de temps plus court que la valeur de temps désirée.

10.6.3.1. SIMATIC-Compteur



Tout comme pour SIMATIC-Temporisation, les compteurs SIMATIC peuvent également être utilisés pour des raisons de compatibilité avec STEP5. Les compteurs sont également globaux.

SIMATIC-Compteurs ont les propriétés systèmes suivantes :

- Ils sont affectés à un nombre fixe,
- Ils ne peuvent pas être utilisés à plusieurs reprises,
- La plage de valeurs est comprise entre 0 et max. 999

Commentaire

Si vous utilisez un compteur SIMATIC, utilisez-le uniquement dans un emplacement du programme, afin d'éviter les erreurs de comptage.

10.6.4. Plage de valeur de différents formats numériques

Format	1 Bit	1 Byte	1 WORD	1 DWORD	1 LWORD
Binary number	2#0 or 2#1	2#0 to 2#1111_ 111	2#0 to 2#1111_1111 _ 1111_1111	2#0 to 2#1111_1111_ 1111_1111_1111_ 1111_1111_1111	2#0 to 2#1111_1111_1111_ 1111_1111_1111_1111_1111_ 1111_1111_1111_1111_1111_ 1111_1111_1111
Octal number	8#0 or 8#1	8#0 to 8#377	8#0 to 8#177_777	8#0 to 8#37_777_777_777	8#0 to 8#1_777_777_ 777_777_777_777_777
Hexadecimal number **(BCD each digit only 0-9)**	16#0 or 16#1	16#0 to 16#FF BCD	16#0 to 16#FFFF	16#0 to 16#FFFF_FFFF	16#0 to 16#FFFF_FFFF_FFFF_FFFF
Decimal number	0 or 1	-128 to +127	-32 768 to +32 767	-2 147 483 648 to +2 147 483 648	-9 223 372 036 854 775 808 to +9 223 372 036 854 775 807
Decimal number without sign	0 or 1	0 to 255	0 to 65 535	0 to 4 294 967 295	0 to 18 446 744 073 709 551 615
Real number (Floating-point number)	/	/	/	-3.402823e+38 to -1.175495e-38 ±0,0 +1.175495e-38 t0 +3.402823e+38	-1.7976931348623158e+308 to -2.2250738585072014e-308 ±0.0 +2.2250738585072014e-308 to +1.7976931348623158e+308



Sommaire 11

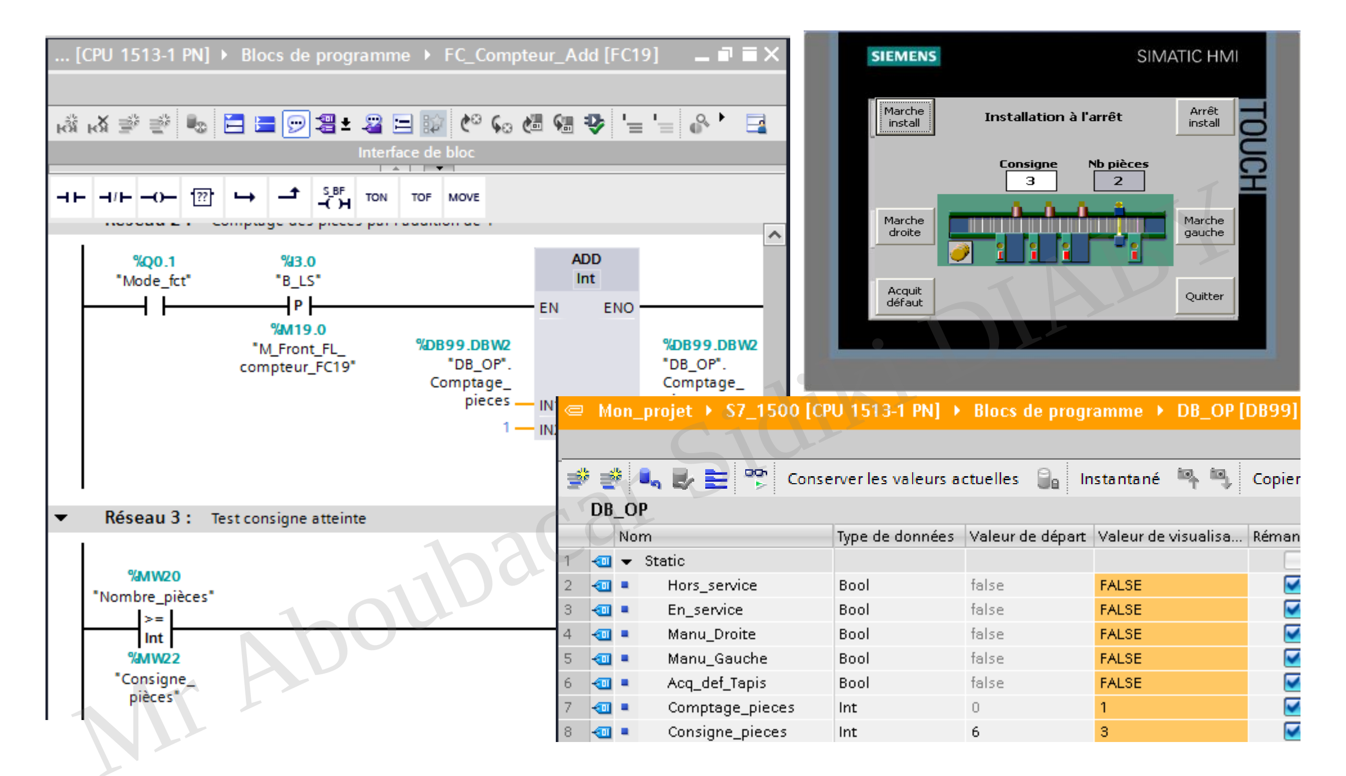
11. Blocs de données

A l'issue du chapitre, vous allez…

- ... connaître l'utilité des blocs de données
- ... connaître les types de données élémentaires et complexes
- ... savoir éditer un bloc de données avec des variables et les utiliser dans un programme
- ... connaître les possibilités d'adressage des variables des blocs de données
- ... savoir lire un bloc de données
- … être capable de faire la différence entre les valeurs par défaut, de départ, de visualisation, et de réglage (consigne)



11.1. Présentation des blocs de données et de leur utilisation



	Nom	Type de données	Valeur de départ	Valeur de visualisa...	Réman
1	Static				
2	Hors_service	Bool	false	FALSE	☑
3	En_service	Bool	false	FALSE	☑
4	Manu_Droite	Bool	false	FALSE	☑
5	Manu_Gauche	Bool	false	FALSE	☑
6	Acq_def_Tapis	Bool	false	FALSE	☑
7	Comptage_pieces	Int	0	1	☑
8	Consigne_pieces	Int	6	3	☑

Généralités

Les blocs de données servent à enregistrer des données utilisateur. Ils occupent de l'espace dans la mémoire utilisateur de la CPU. Les blocs de données contiennent des variables (par exemple des valeurs numériques) nécessaires à l'exécution du programme utilisateur.

Le programme utilisateur peut accéder aux données d'un bloc de données par des opérations sur bit, sur octet, sur mot ou sur double mot. L'accès peut se faire en symbolique ou en absolu.

Domaine d'application

Les blocs de données peuvent, selon leur contenu, être employés de différentes manières par l'utilisateur. Il convient de distinguer :

- les blocs de données globaux : ils contiennent des informations accessibles à partir de tous les blocs de code du programme utilisateur. Les blocs de données globaux servent souvent également d'interface avec les pupitres HMI (voir ci-dessus).
- les blocs de données d'instance : ils sont toujours associés à un bloc fonctionnel (FB). Les données de ce DB ne devraient être traitées que par le FB correspondant. Les blocs de données d'instance sont traités plus en détail au chapitre « Fonctions et blocs fonctionnels ».

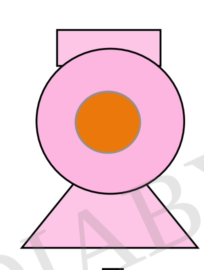
Création de DB

Les DB globaux sont créés avec l'éditeur de programme ou à partir d'un « type de données utilisateur » qui aura été créé préalablement. Les blocs de données d'instance sont créés en cas d'appel d'un bloc fonctionnel (FB).

Registres

La CPU possède deux registres de blocs de données, les registres DB et DI, ce qui permet d'ouvrir deux blocs de données simultanément.

11.2. Signification des variables et des types de données



Les types de données définissent les propriétés intrinsèques des données.

- Plage continue : par ex. la vitesse réelle de rotation du moteur
- Propriété de type « oui/non » : par ex. présence d'un défaut

Le types de données permettent de définir :

- la plage de valeurs possible (INT : - 32 768 ... + 32 767, etc.)
- les opérations admissibles (les opérations arithmétiques : +, -, etc.)
- Les types de données font abstraction de la représentation binaire correspondante dans la mémoire.

Les variables permettent de mémoriser des valeurs pour un traitement ultérieur.

Vitesse réelle : REAL

Vitesse de consigne : REAL

Défaut : BOOL

Validation : BOOL

Opérandes « classiques » pour les automates programmables

En programmation classique, l'accès aux adresses mémoire de l'API s'effectue directement en saisissant le type d'opérande (par ex. : M=Mémento, I=Entrée, etc.), la longueur d'accès (par ex.: B=Octet, W=Mot, etc.) et l'adresse binaire (octet/bit). L'accès en adressage absolu à ces zones mémoire peut être utilisé à différentes fins et avec différents formats à l'intérieur du programme, par ex. sous forme d'entier double (par ex. DINT), de nombre réel à virgule flottante (par ex. REAL) ou dans un format regroupant différents signaux (par ex. WORD). Jusqu'à présent, il appartenait à l'automaticien en charge de la programmation de retenir le format et la finalité des différentes zones mémoire, avec les risques d'erreurs que cela pouvait comporter pour la programmation.

Signification des variables

Les variables sont, avec les instructions, les éléments les plus importants en programmation. En effet, il faut souvent enregistrer des valeurs dans le programme pour pouvoir les traiter ultérieurement. La valeur d'une variable peut ainsi être mémorisée à un endroit « quelconque » dans la mémoire de l'API.

Les données représentent une abstraction de la réalité. A ce titre, elles ne tiennent pas compte des propriétés concrètes des objets qu'elles représentent.

Type de données

Le choix de la représentation des données est souvent difficile. Ce choix est en outre souvent limité par les possibilités disponibles. En effet, le type de données choisi doit, d'une part, restituer correctement les propriétés des objets représentés par les données, et d'autre part, permettre l'exécution des opérations requises pour la commande du processus.

Le type de données détermine les valeurs qui pourront être traitées et les opérations qui pourront être exécutées. Le type de données définit ainsi de manière univoque :

- la plage de valeurs possible
- les opérations admissibles
- comment le modèle de bits doit être interprété

11.2.1. Types de données dans STEP 7



Types de données élémentaires

Les types de données élémentaires sont définis par la norme CEI 61131-3. Ces types de données ont toujours une longueur inférieure ou égale à 64 bits. Ainsi, ils peuvent être chargés entièrement dans les accumulateurs du processeur S7 et traités par les instructions élémentaires de STEP 7.

Types de données complexes

Les types de données complexes sont constitués d’une structure de données comprenant des types de données élémentaires et/ou complexes. Les types de données complexes sont utilisés uniquement pour la déclaration de variables dans des blocs de données globaux et, à l’intérieur des blocs, pour la déclaration de variables locales (TEMP, STAT) ou de paramètres (IN, OUT et INOUT). Les variables définies avec des types de données complexes ne peuvent pas être entièrement traitées avec des opérations élémentaires.

11.2.1.1. Types de données complexes 1

Type de données	Longueur (Bits)	S7-1200	S7-1500	Exemple
DT (DATE_AND_TIME)	**64**	✗	✓	**DT#2008-10-25-08:12:34.567**
DTL	**96**	✓	✓	**DTL#1976-12-16-20:30:20.250**
STRING	**8 * (Nombre de caractères+2)**	✓	✓	**´C'est une chaine´** max. 254 caractères en format ASCII
WSTRING (Wide Character String)	**16 * (Nombre de caractères+2)**	✓	✓	**WSTRING#´STRING dans le format UNICODE** Supérieur à 16382 caractères dans le format unicode

DT

Le type de données DATE_AND_TIME représente un moment dans le temps constitué de la date et de l'heure du jour. Au lieu de DATE_AND_TIME, l'abréviation DT peut également être utilisée.

Octet	Details	Palge de valeurs
0	Année	0 to 99 (Years 1990 to 2089) BCD#90 = 1990 … BCD#0 = 2000 … BCD#89 = 2089
1	Mois	BCD#0 to BCD#12
2	Jour	BCD#0 to BCD#31
3	Heure	BCD#0 to BCD#23
4	Minute	BCD#0 to BCD#59
5	Seconde	BCD#0 to BCD#59
6	Les deux chiffres les plus significatifs de MSEC	BCD#0 to BCD#999
7 (4MSB) 1)	Le chiffre le moins significatif de MSEC	BCD#0 to BCD#9
7 (4MSB) 2)	Jour de la semaine	BCD#1 to BCD#7 BCD#1 = Sunday … BCD#7 = Saturday

DTL

Le type de données DTL a une longueur de 12 octets et, comme LDT, stocke des informations sur la date et l'heure précise à la nanoseconde depuis le 1.1.1970, seulement dans une structure prédéfinie.

Avantage: les valeurs individuelles (jour, heure, etc.) sont plus faciles à lire.

Octet	Details	Type	Plage de valerus
0 - 1	Année	UINT	1970 to 2554
2	Mois	USINT	1 to 12
3	Jour	USINT	1 to 31
4	Jour de la semaine	USINT	1 (Sunday) to 7 (Saturday) The weekday is not considered in the value entry.
5	Heure	USINT	0 to 23
6	Minute	USINT	0 to 59
7	Seconde	USINT	0 to 59
8 -11	Nanoseconde	UDINT	0 to 999 999 999

STRING

Le type de données STRING stocke plusieurs caractères ASCII du type de données CHAR dans une chaîne de caractères d'un maximum de 254 caractères.

WSTRING

Le type de données WSTRING (Wide Character String) stocke plusieurs caractères Unicode du type de données WCHAR dans une chaîne de caractères d'un maximum de 16382 caractères.

11.2.1.2. Types de données complexes 2

Type de données	Longueur (Bits)	S7-1200	S7-1500	Exemple
ARRAY	Définie par l'utilisateur	✓	✓	Valeurs de mesures : ARRAY[1..20] of INT;
STRUCT	Définie par l'utilisateur	✓	✓	Moteur: STRUCT Vitesse : REAL; Consigne : UINT; Erreur : BOOL; Activation : BOOL; END_STRUCT
WSTRING (Wide Character String)	Définie par l'utilisateur	✓	✓	**Définie** (dans le dossier "Type de données utils.") PLC-DT1 : STRUCT vitesse : REAL; Activation: BOOL; Defaut: BOOL; END_STRUCT **Utilisation** (dans les blocs de données et interface de blocs) Moteur1 : PLC-DT1; Moteur2 : PLC-DT1; Moteur3 : PLC-DT1; :

Les tableaux et les structures sont constitués de groupes de types de données élémentaires ou complexes.

Ils vous permettent de créer des types de données adaptés à votre tâche avec lesquels vous pouvez structurer de grandes quantités de données et les traiter symboliquement.

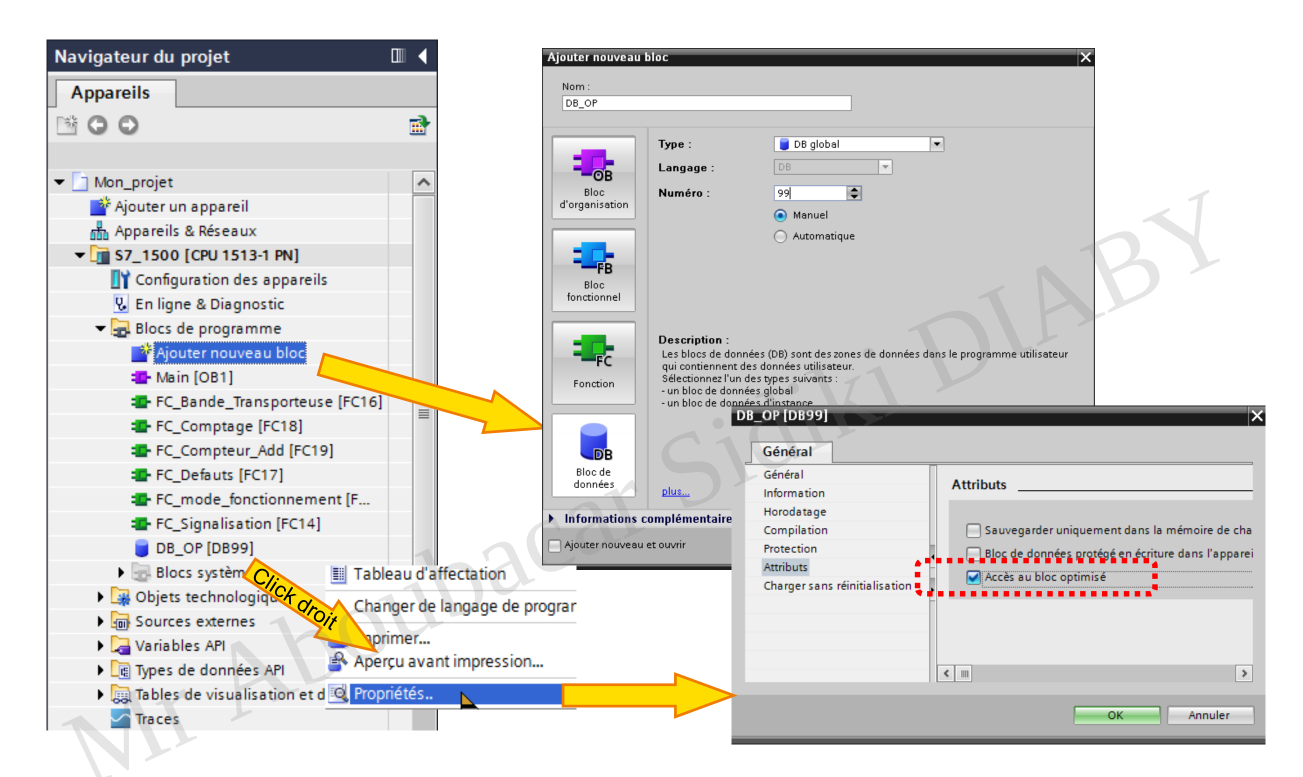
ARRAY e STRUCT

Les tableaux et les Structures ne peuvent être déclarés que dans des blocs de données globaux et comme des paramètres ou des variables locales de blocs de code.

PLC- type de donnée (Type de donnée défini par l'utilisateur)

Les types de données définis par l'utilisateur représentent des structures auto-définies. Cette structure est stockée dans les types de données de l'automate et peut être utilisée comme "modèle" dans le type de données d'une autre variable.

11.3. Créer un bloc de données global :



Créer un bloc de données global

La création de blocs de données globaux s'effectue de la même manière que pour les blocs de code.

Lors de la création d'un bloc de données, vous devez définir le type du bloc de données ou son usage :

- DB globaux pour la mémorisation de données globales ou la création de variables globales
- DB d'instance ou « mémoire privée » pour les blocs fonctionnels utilisateur ou une « instruction » spéciale derrière laquelle se cache un bloc fonctionnel (FB)

Attribut « Accès au bloc optimisé »

Sur le S7-1200/1500, les blocs de données peuvent être créés avec l'attribut « Accès au bloc optimisé » :

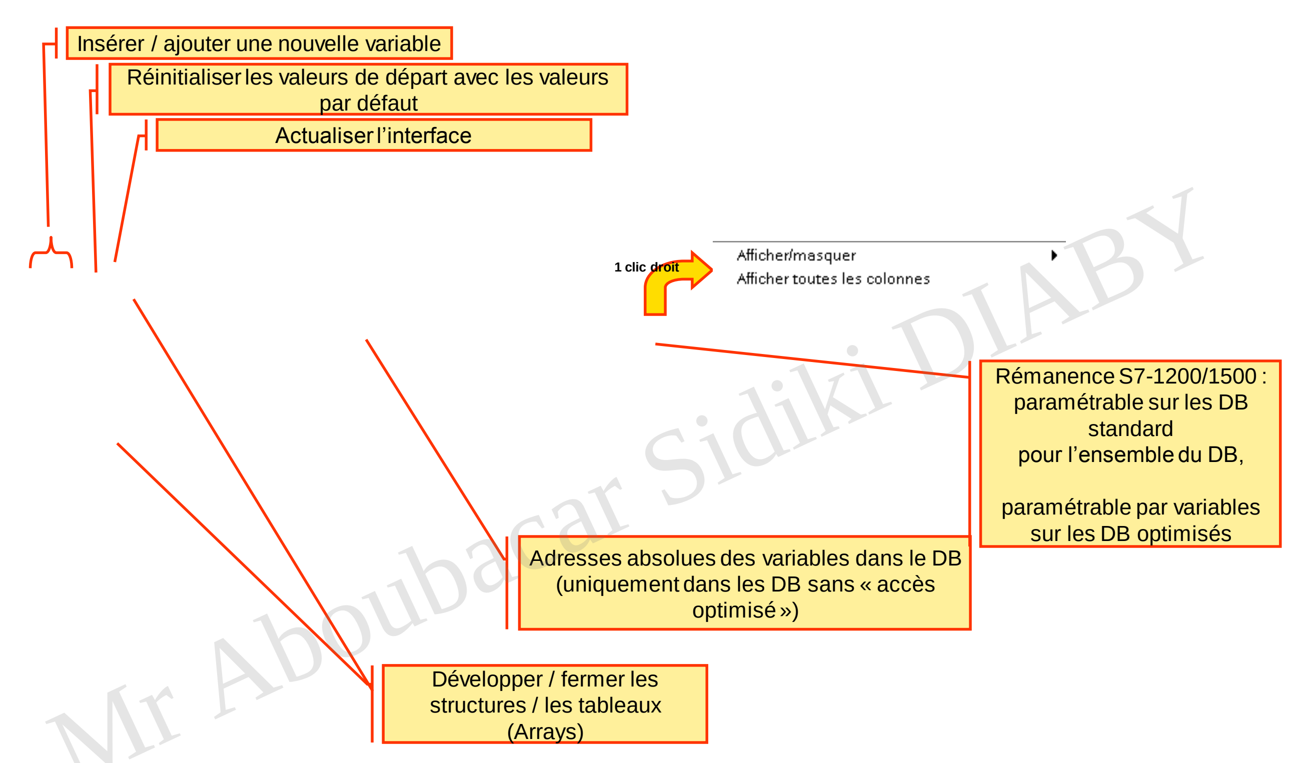
- sur le S7-1200, ces blocs de données sont créés avec une optimisation mémoire. Ils ne font plus appel aux octets de remplissage utilisés par les S7-300/400 pour la création successive de variables de formats bit, octet, mot et double-mot, ce qui libère de la place en mémoire.
- sur le S7-1500, ces blocs de données sont créés avec une optimisation en accès compte tenu du code machine S7, ce qui minimise les temps d'accès aux variables.

Accès au bloc optimisé voir: Centre d'information du portail TIA> Documentation> Manuels> Technique de commande> Guide de programmation pour une programmation optimale des automates SIMATIC S7-1200 / 1500

Attribut « Sauvegarder uniquement dans la mémoire de chargement » (sur S7-300/400 Attribut « unlinked »)

signifie que le bloc de données est uniquement chargé dans la mémoire de chargement de la CPU (protection en écriture) lors du chargement dans la CPU et pas automatiquement chargé dans la mémoire de travail à partir de là.

11.3.1. Editeur de blocs de données



Décalage

Le décalage désigne l'adresse absolue d'une variable au sein d'un DB. Dans le programme STEP7, il est cependant préférable d'utiliser l'adresse ou le nom symboliques, car ils sont plus lisibles et moins sources d'erreurs que l'adresse absolue.

Sur le S7-300/400, l'adresse absolue des variables est toujours affichée au sein du bloc. Sur les S7-1200 et S7-1500, elle n'est affichée que si le bloc de données a été créé sans l'option « Accès aux blocs optimisé ».

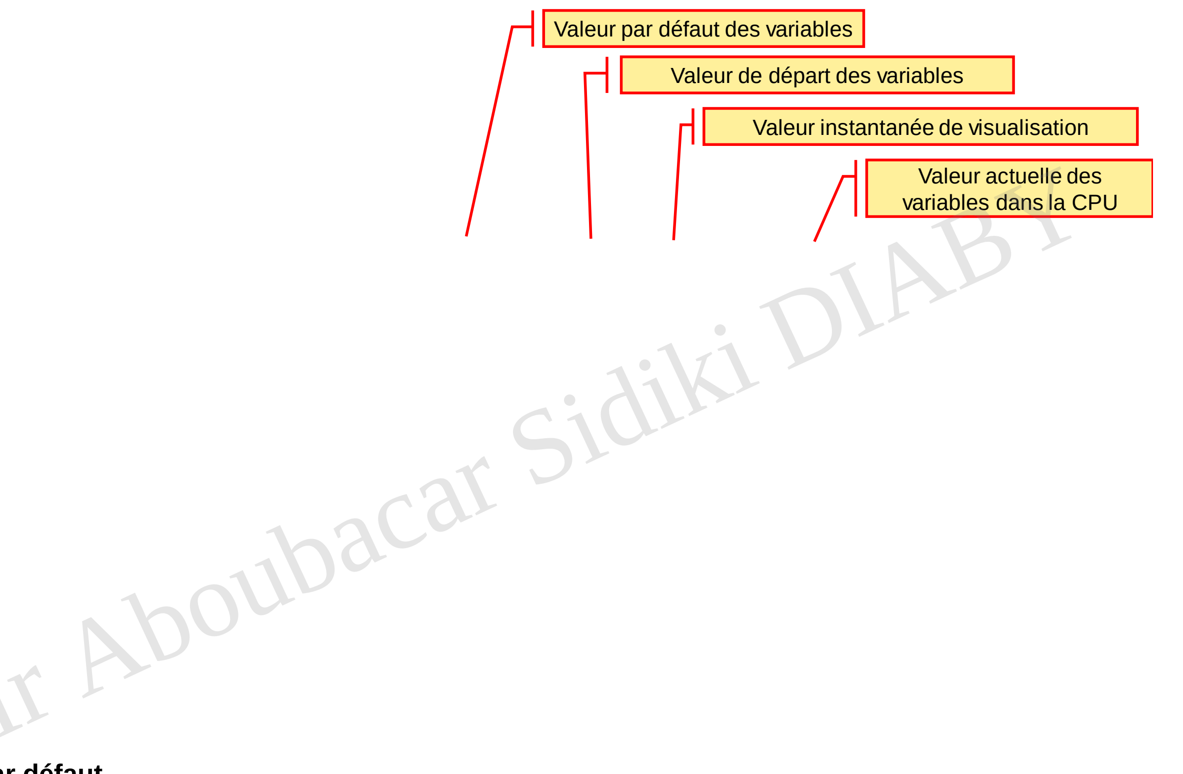
Rémanence

Sur le S7-300/400, le comportement rémanent ne peut pas être défini individuellement pour les différentes variables. La définition de la rémanence s'applique toujours à l'ensemble des variables ou du bloc de données.

Si la « Rémanence » est activée, les valeurs de visualisation sont conservées jusqu'à un nouveau chargement du bloc de données dans la CPU.

Si la « Rémanence » n'est pas activée, les valeurs de visualisation sont écrasées dans la mémoire de travail par les valeurs de départ de la mémoire de chargement à chaque passage de STOP -> RUN de la CPU (démarrage à froid, redémarrage, démarrage par PG, sélecteur de mode ou mise hors tension->sous tension).

11.3.2. Valeur par défaut, valeur de départ, valeur de visualisation



Valeur par défaut

Les valeurs par défaut ne sont pas éditables au sein des blocs de données globaux. Elles le sont uniquement au sein des types de données API (UDT), au niveau des valeurs par défaut des éléments de structure, et au sein des FB, au niveau des valeurs par défaut des paramètres et variables statiques.

Si une variable du type de donnée Structure selon type de données API (UDT) est déclarée au sein d'un DB global, les valeurs par défaut éditées dans UDT sont alors affichées dans le bloc de données (non éditable) et reprises comme valeurs de départ (éditables).

Au sein d'un FB, il faut également spécifier les valeurs par défaut des paramètres et des variables statiques lors de leur déclaration. Ces valeurs par défaut (non éditables) sont alors affichées dans les DB d'instance et reprises comme valeurs de départ (éditables).

Valeur de départ

Lors de la déclaration d'une variable, sa valeur par défaut est automatiquement reprise comme valeur de départ (éditable). Après le premier chargement du DB dans la CPU, la CPU commence le traitement du programme avec cette valeur. En cas de nouveau chargement du DB dans la CPU, le comportement de la CPU dépend du paramétrage de la rémanence des variables DB et du type de chargement du DB (avec ou sans réinitialisation, voir chapitre « Programmation avancée ».

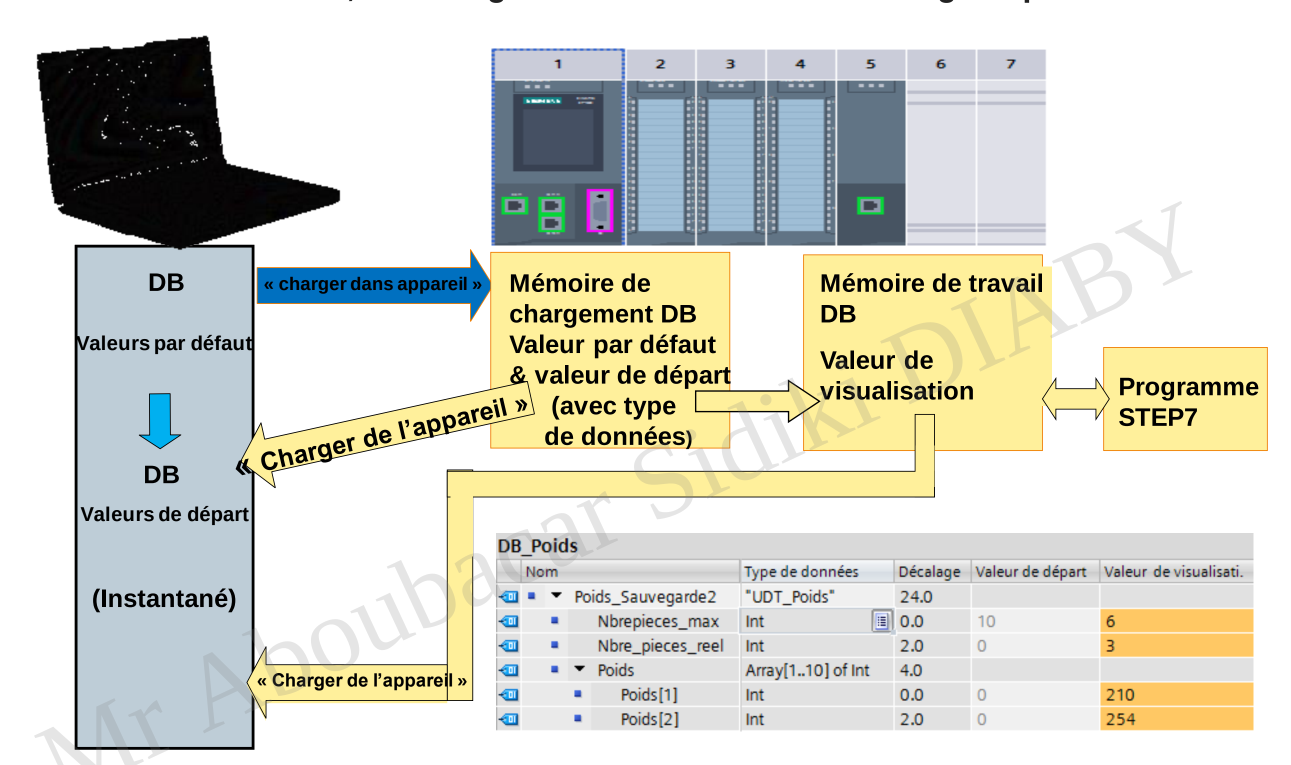
Valeur de visualisation

La valeur de visualisation d'une variable est la valeur courante que la variable a actuellement dans la mémoire de données de la CPU.

Instantané

La valeur « Instantané » est une valeur de visualisation à un instant x déjà passé, au cours duquel les valeurs de visualisation instantanées ont été lues dans la CPU et mémorisées hors ligne sous forme de valeurs d'instantanés.

11.3.3. Remmanence, télécharger DB dans la CPU / télécharger à partir de la CPU



Nom	Type de données	Décalage	Valeur de départ	Valeur de visualisati.
Poids_Sauvegarde2	"UDT_Poids"	24.0		
Nbrepieces_max	Int	0.0	10	6
Nbre_pieces_reel	Int	2.0	0	3
Poids	Array[1..10] of Int	4.0		
Poids[1]	Int	0.0	0	210
Poids[2]	Int	2.0	0	254

« Charger dans l'appareil » (Download)

Lors du chargement d'un bloc de données de la PG dans la CPU, les valeurs par défaut et de départ ainsi que les types de données des variables sont transmis dans la mémoire de chargement de la CPU. A partir de la mémoire de chargement, les valeurs de départ sont ensuite reprises automatiquement dans la mémoire de travail de la CPU comme valeurs de visualisation. Le programme STEP 7 travaille avec les valeurs de visualisation de la mémoire de travail.

« Charger depuis l'appareil » (Upload)

Lors du chargement (upload), les blocs de données dont les valeurs de départ sont différentes en ligne et hors ligne sont créés dans le projet hors ligne sous un autre nom en tant que nouveaux blocs de données supplémentaires. Les blocs de données déjà existants dans le projet hors ligne ne sont donc pas écrasés.

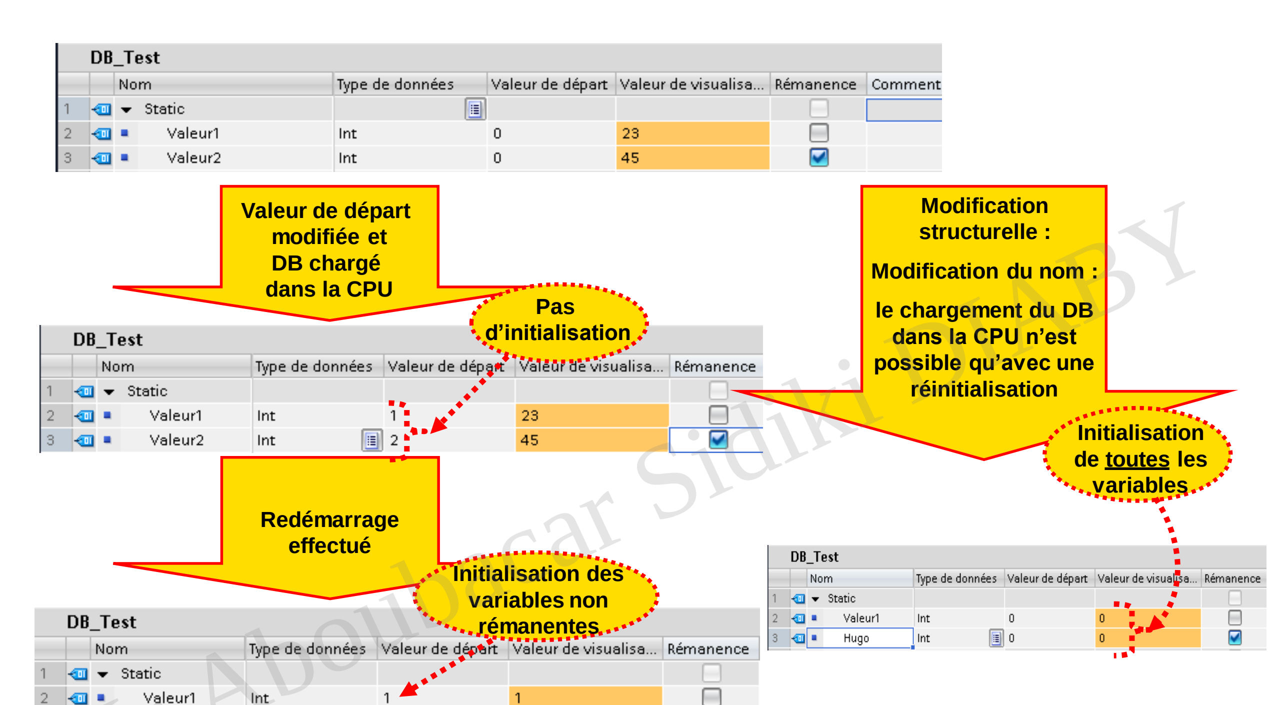
Dans ces blocs de données créés en supplément, les valeurs par défaut chargées à partir de la CPU sont également reprises hors ligne comme valeurs par défaut et les valeurs de visualisation chargées à partir de la CPU sont reprises hors ligne comme instantanés et valeurs de départ.

Rémanence (Retain)

Avec les variables rémanentes, les "valeurs de surveillance" (valeurs réelles) sont conservées après un redémarrage de la CPU.

Les variables non rémanentes occupent la mémoire de travail mais pas de mémoire rémanente et sont ainsi réinitialisées aux valeurs de démarrage à partir de la mémoire de chargement à chaque redémarrage de l'UC (Power Off/On ou après chaque STOP - RUN).

11.3.3.1. Téléchargement de blocs de données modifiés dans le processeur



Chargement de blocs de données avec réinitialisation

Une modification des valeurs initiales ne constitue pas une modification structurelle, que le DB ait été créé avec un accès au bloc standard ou optimisé. C'est pourquoi elle peut être réalisée et chargée dans la CPU sans qu'une réinitialisation des variables du bloc de données soit nécessaire.

Après des modifications structurelles en revanche, le chargement du bloc de données n'est possible qu'avec une réinitialisation de toutes les variables.

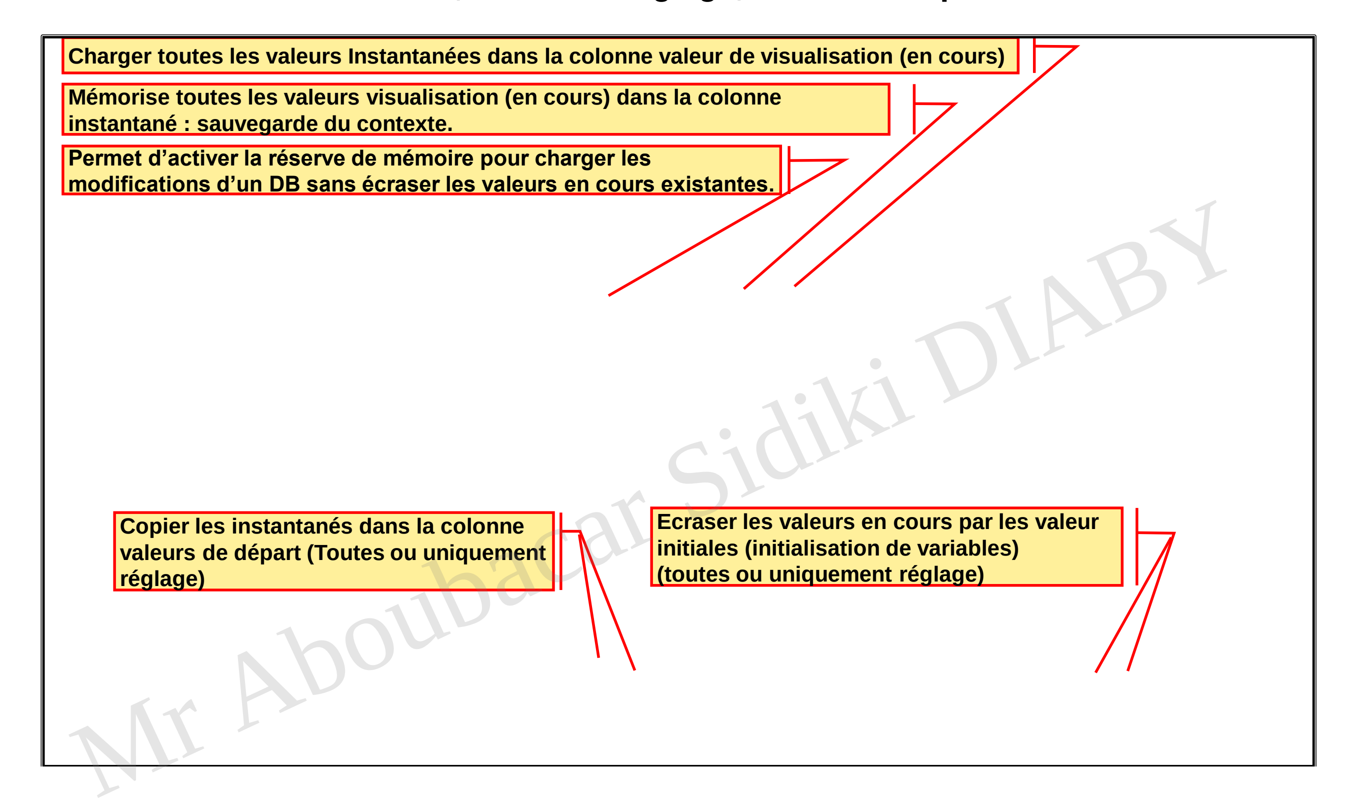
Les modifications structurelles sont :

- les modifications de noms
- les modifications du comportement rémanent
- l'adjonction / la suppression de variables

Réinitialisation du DB en cas de redémarrage

Qu'un DB ait été créé avec un accès au bloc standard ou optimisé, toutes les valeurs de visualisation (valeurs actuelles) des variables non rémanentes sont remises à leur valeur initiale en cas de redémarrage de la CPU.

11.3.4. Valeur instantanée, valeur de réglage, valeur de départ



Instantané

Avec la fonction " Instantané de la valeur du moniteur ", les valeurs réelles du DB en ligne sont stockées dans le DB hors ligne.

Avec la fonction "Charger l'instantané en tant que valeur de surveillance / valeur réelle", les valeurs sont à nouveau chargées dans le DB en ligne.

Copier l'instantané pour démarrer les valeurs

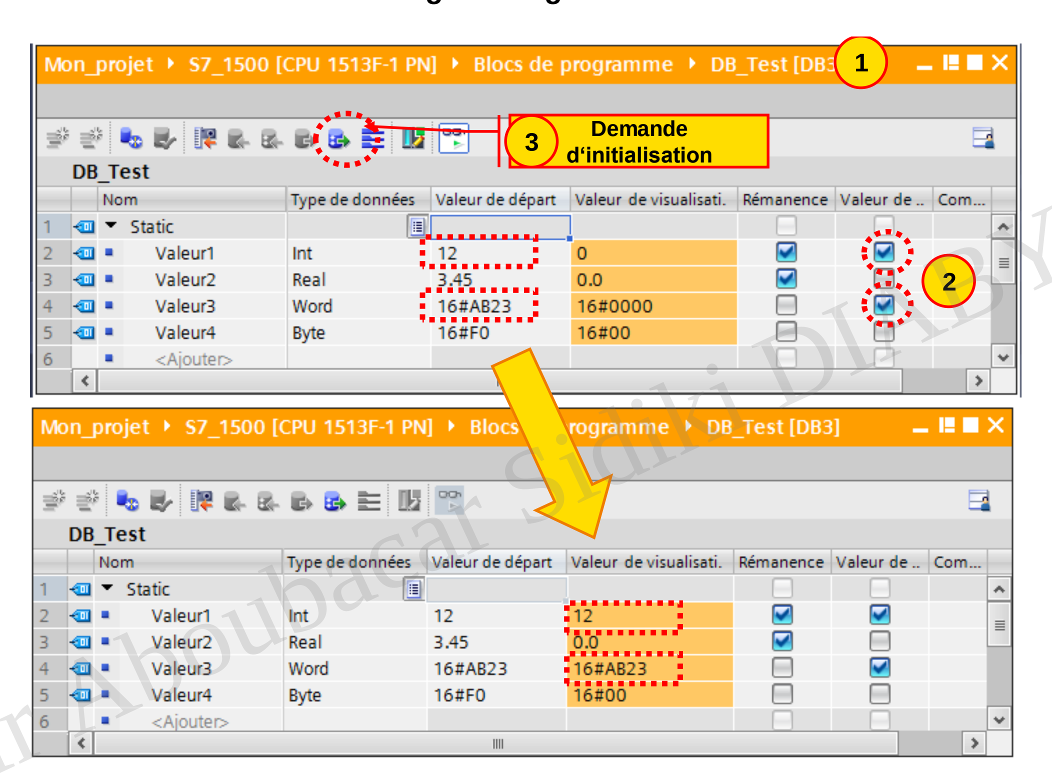
Les valeurs sauvegardées peuvent être adoptées dans la colonne Valeur de départ au moyen du bouton (pour tous les instantanés sélectionnés dans la colonne Point de consigne) ou du bouton (pour tous les instantanés).

La prochaine fois que ce DB est transféré, ces valeurs sont appliquées comme valeur de départ (pas de changement structurel de la BD -> un téléchargement sans ré-initialisation est possible).

Ajustement des valeurs réelles

De plus, la totalité du DB peut être initialisée avec le bouton et toutes les valeurs sélectionnées dans la colonne Setpoint peuvent être initialisées avec le bouton .

11.3.4.1. Initialiser les valeurs de consigne en ligne



Initialiser les valeurs de réglages dans un programme en ligne

Toutes les variables sélectionnées en « Valeur de réglage », peuvent être initialisées dans la CPU en ligne. Les valeurs de visualisation en ligne seront écrasées par les valeurs de départ éditables hors ligne. La CPU reste dans l'état « RUN ». Les valeurs de visualisation modifiées (valeurs actuelles) seront prises en compte globalement lors du prochain contrôle de cycle. Ceci s'applique aussi bien pour des variables rémanentes ou non rémanentes. Le traitement du programme continue avec les nouvelles valeurs des variables. Les pré-requis sont les suivants une connexion en ligne avec la CPU, une structure identique du bloc de données en ligne et hors ligne et la ou plusieurs variables sélectionnées comme « Valeur de réglage »

Marche à suivre:

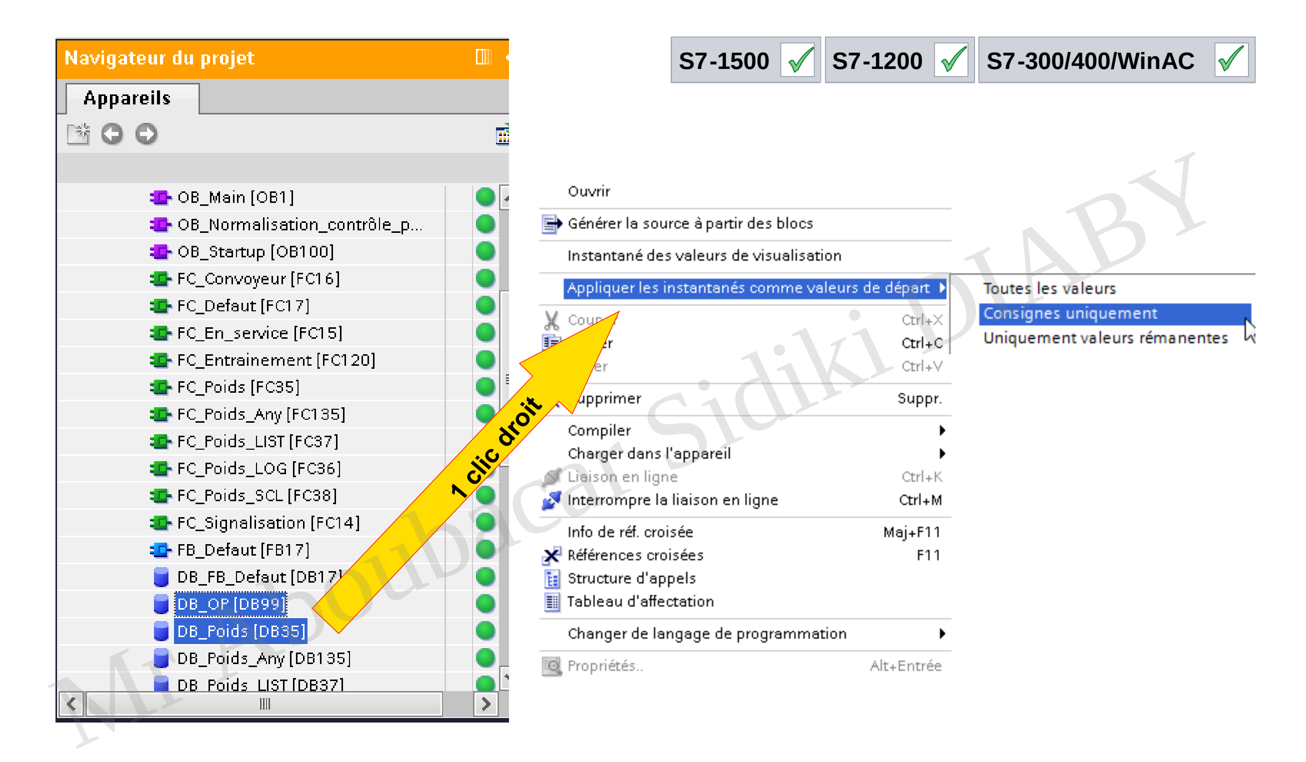
1. Ouvrez le bloc de données global ou de bloc de données d'instance et réaliser une liaison en ligne.
2. Sélectionnez dans la colonne « Valeur de réglage » les variables pour lesquelles les valeurs actuelles dans la CPU doivent être écrasées avec les valeurs de départ.
3. Cliquez sur le bouton « Initialiser les valeurs de consignes », pour initialiser les variables sélectionnées sous « Valeur de réglage »

Note:

Pour les blocs de données globaux « initialiser les valeurs de consigne » peut être sélectionné et désélectionné sans avoir à recharger le bloc uniquement pour les variables qui ne sont pas du type de données API « UDT ». Pour les variables de type de données API « UDT » il faut sélectionner « Valeur de réglage » dans l'UDT puis recharger le DB.

Pour les DB d'instance, seuls les variables statiques peuvent être sélectionnées par « Initialiser les valeurs de consigne ».

11.3.4.2. Modification de l'instantané / valeur de début de plusieurs / tous les blocs de données



Instantané de plusieurs / tous les blocs de données

Tout comme un instantané peut être fait, réécrit, la base de données initialisée et les instantanés copiés dans les valeurs de départ pour un bloc de données individuel, cela peut également être fait pour plusieurs ou tous les blocs.

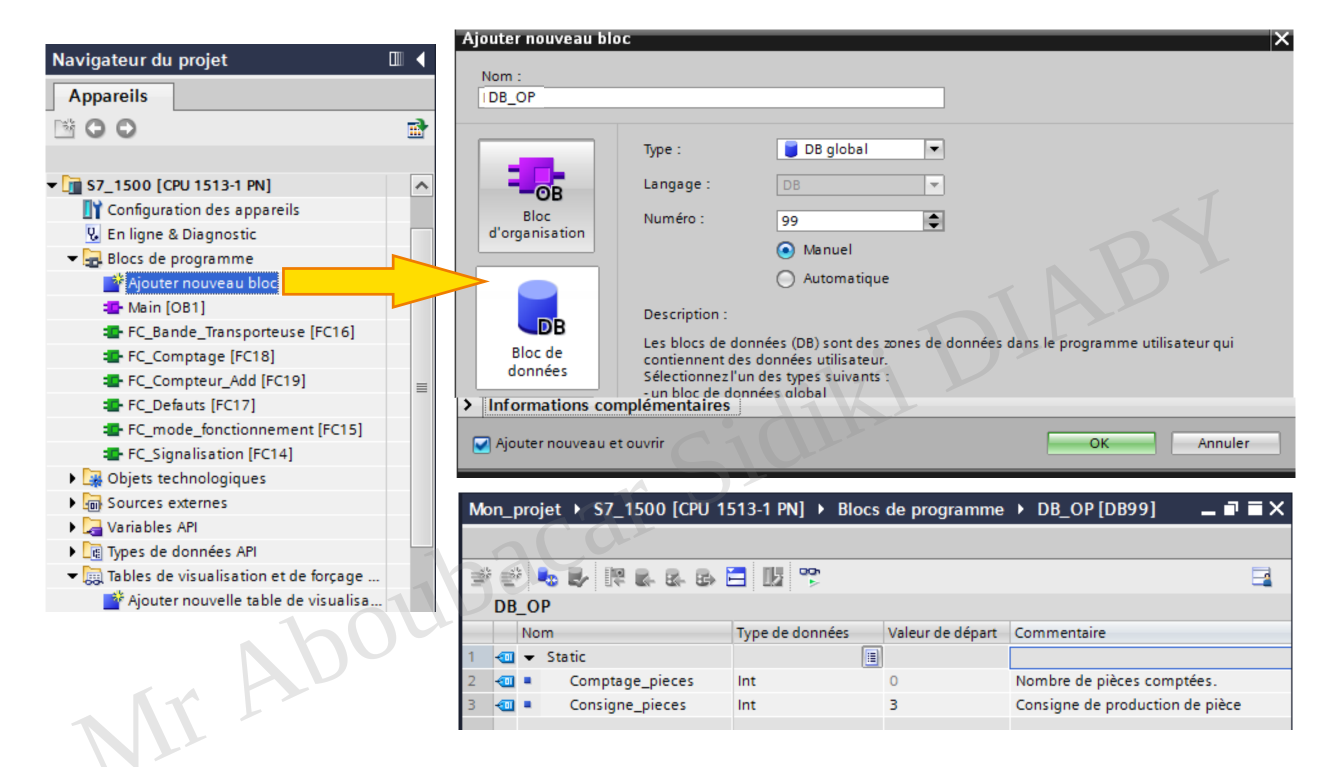
11.3.5. Copier-coller dans / depuis Microsoft Excel



Copier&Coller dans et depuis Excel

La fonction copier&coller dans Windows permet de copier-coller facilement vers Excel, et inversement, une ou plusieurs variables sélectionnées dans une table de variables API ou un bloc de données pour pouvoir les traiter.

11.4. Exercice 1 : Création du bloc de données « DB_OP » (DB99)



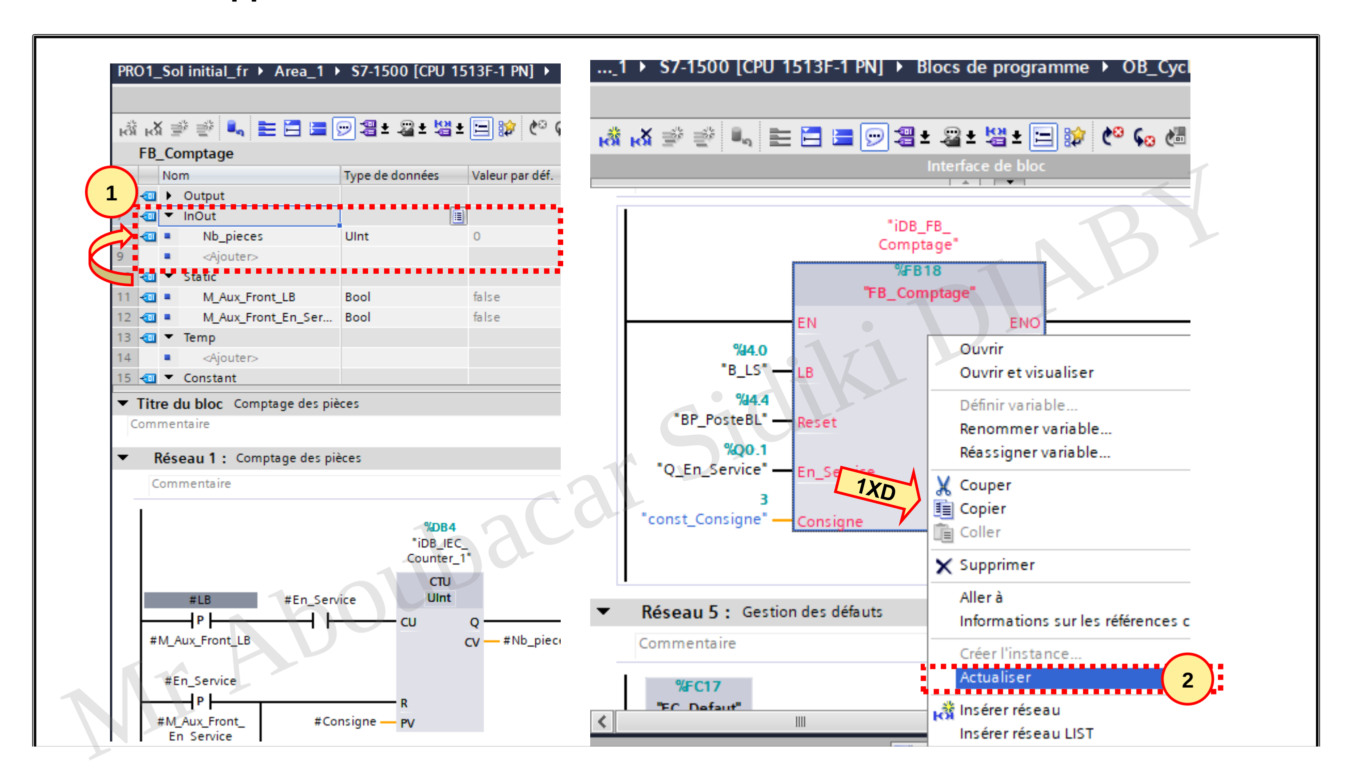
Enoncé :

Il s'agit de créer le bloc de données « DP_OP » (DB99) avec les variables indiquées ci-dessus, qui serviront ensuite d'interface avec le pupitre opérateur.

Marche à suivre :

1. Créez le nouveau bloc « DB_OP » (DB99) en tant que DB global.
2. Déclarez les variables comme indiqué ci-dessus.
3. Enregistrez le projet.

11.4.1. Exercice 2: évolution de « FB_Comptage » et mise à jour de l'appel



Fonction actuelle:

Les pièces transportées sont comptées dans "FB_ Comptage " dès qu'elles ont franchi la barrière lumineuse.

Le nombre actuel, c'est-à-dire la quantité réelle, est stocké dans la variable statique #Act et la quantité de consigne est préréglée avec la constante 3 dans le programme.

Si la quantité réelle a atteint la quantité de consigne, elle est indiquée sur le voyant du modèle de convoyeur " Q_PosteLS " (Q3.4) avec un clignotement de 1Hz et aucune nouvelle séquence de transport ne peut être démarrée. En acquittant ce signal avec le bouton-poussoir " BP_PosteBL " du convoyeur (I 3.4), la variable statique # Nb_pieces est remplacée par 0 et d'autres séquences de transport peuvent être démarrées.

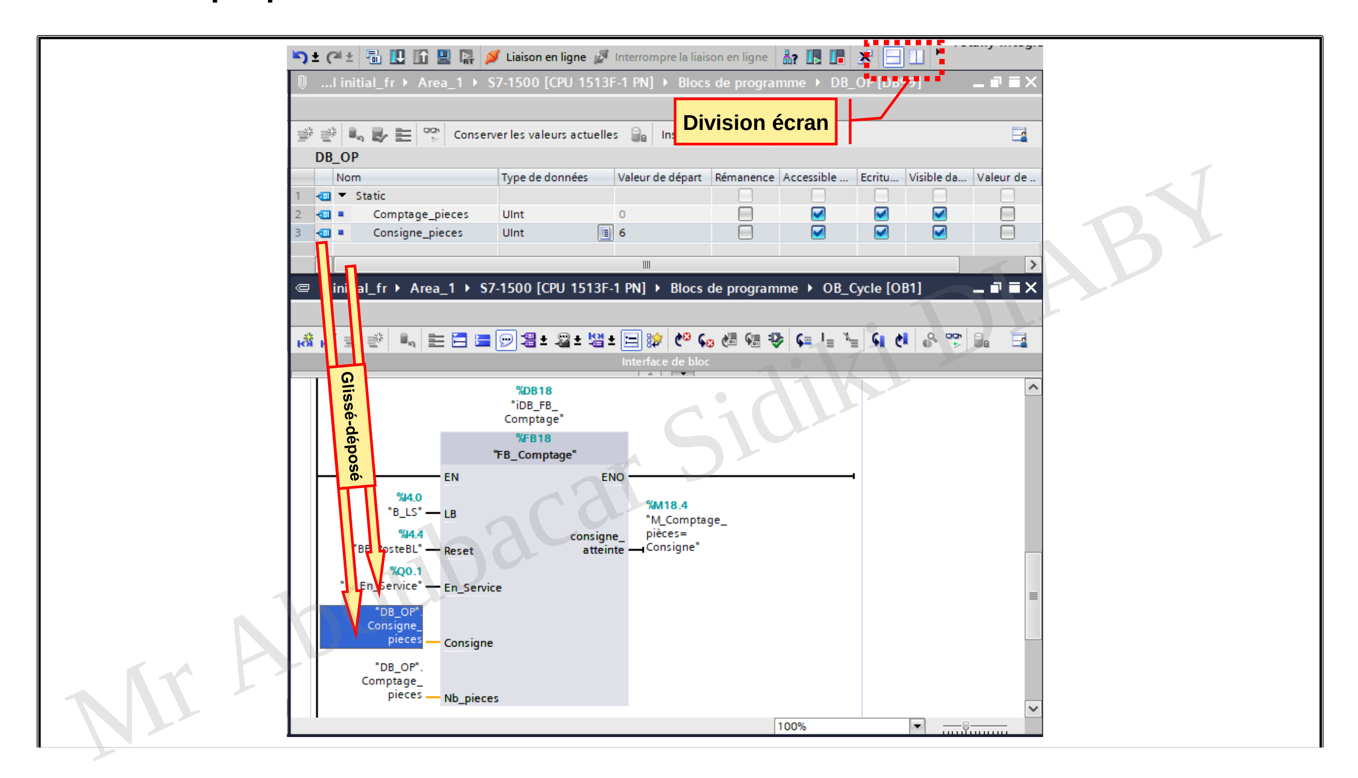
Enoncé

La fonction de « FB_Comptage » reste inchangée, cependant, la quantité réelle ne doit plus être stockée dans la variable statique # Nb_pieces et la quantité de consigne ne doit plus être préréglée avec la constante 3. Au lieu de cela, les variables de bloc de données " DB_OP ". Comptage_pieces et" DB_OP ". Consigne_pieces doivent être utilisés.

Marche à suivre :

1. Déplacez la variable statique # Nb_pieces et posez la dans un paramètre InOut du même type de données (UINT), enregistrez le bloc de fonction.
2. Ouvrez le bloc Organisation "OB_Cyclique".
3. Mettez à jour l'appel de "FB_Comptage". Pour cela, ouvrez le menu contextuel en cliquant avec le bouton droit sur l'appel de "FB_ Comptage " et en activant la fonction "Mettre à jour l'appel de bloc".
4. Confirmez la boîte de dialogue "Synchronisation de l'interface" avec "OK".
5. Renseignez les paramètres formels.

11.4.2. Exercice 3: Utilisation de variables de base de données en tant que paramètres réels



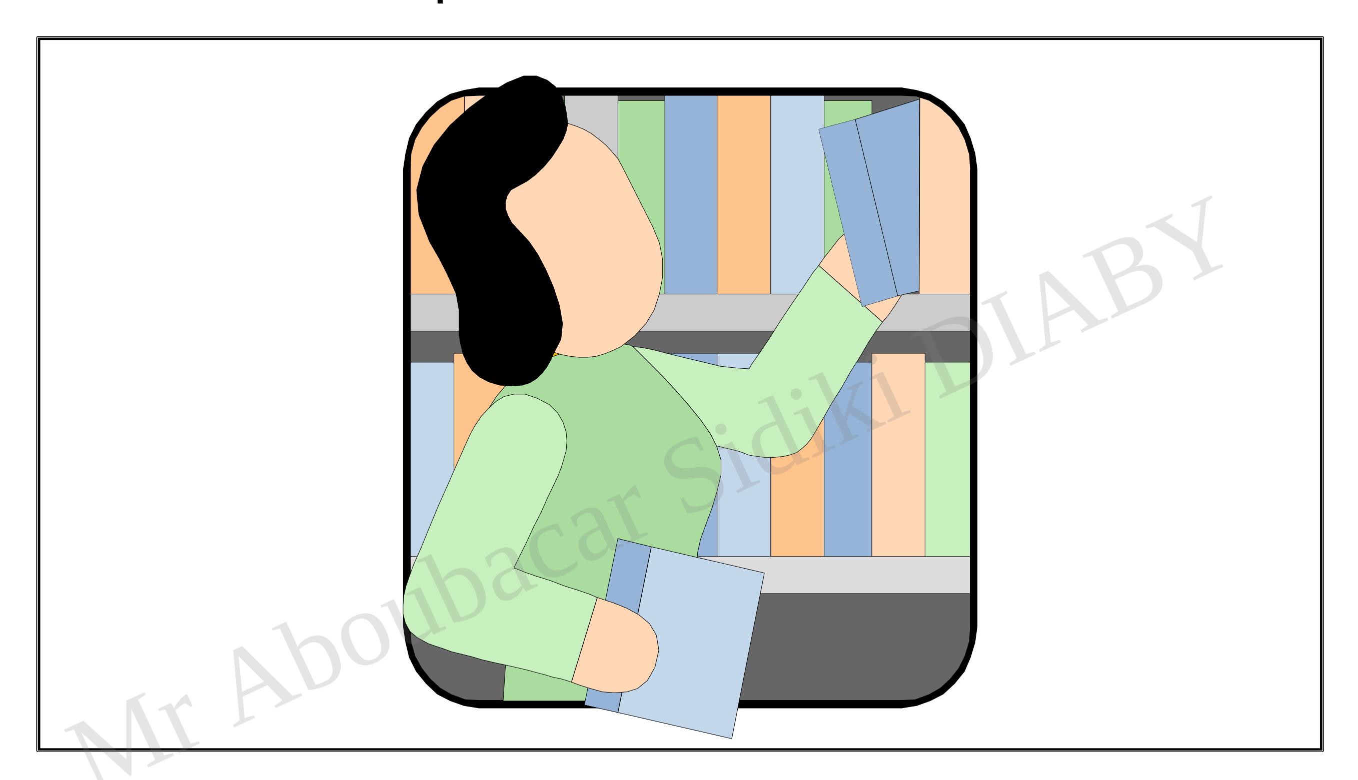
Enoncé :

Vous devez fournir les paramètres formels #Consigne et #Nb_pieces avec les variables de bloc de données " Comptage_pieces " et " Consigne_pieces " du bloc de données "DB_OP".

Marche à suivre :

1. Divisez la zone de travail à l'aide du bouton "Split Editor".
2. Dans une zone, ouvrez le bloc de données "DB_OP" et dans l'autre le bloc d'organisation "OB_Cyclique".
3. Fournissez les paramètres formels " Comptage_pieces " et " Consigne_pieces " avec les variables de bloc de données "Consigne_pieces" et "Comptage_pieces" du bloc de données "DB_OP" en utilisant le glisser-déposer.
4. Compilez et enregistrez votre projet.
5. Visualisez le bloc de données "DB_OP" pendant le transport des pièces en mode automatique.

11.5. Pour en savoir plus



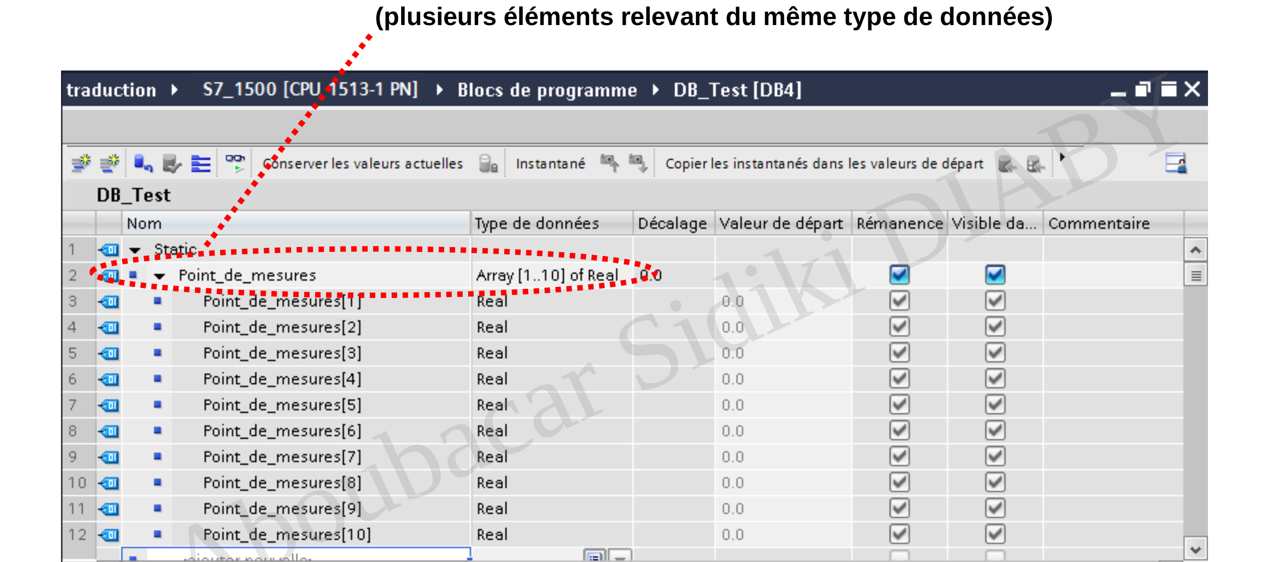
Commentaire

Les pages suivantes contiennent d'autres informations ou sont destinées à compléter un sujet.
Pour une étude plus approfondie, nous proposons des cours avancés et des supports d'auto-apprentissage.

Mr Aboubacar Sidiki DIABY

11.5.1. Exemple de variable de type ARRAY

Tableau portant le nom « Point_de_mesures »
(plusieurs éléments relevant du même type de données)



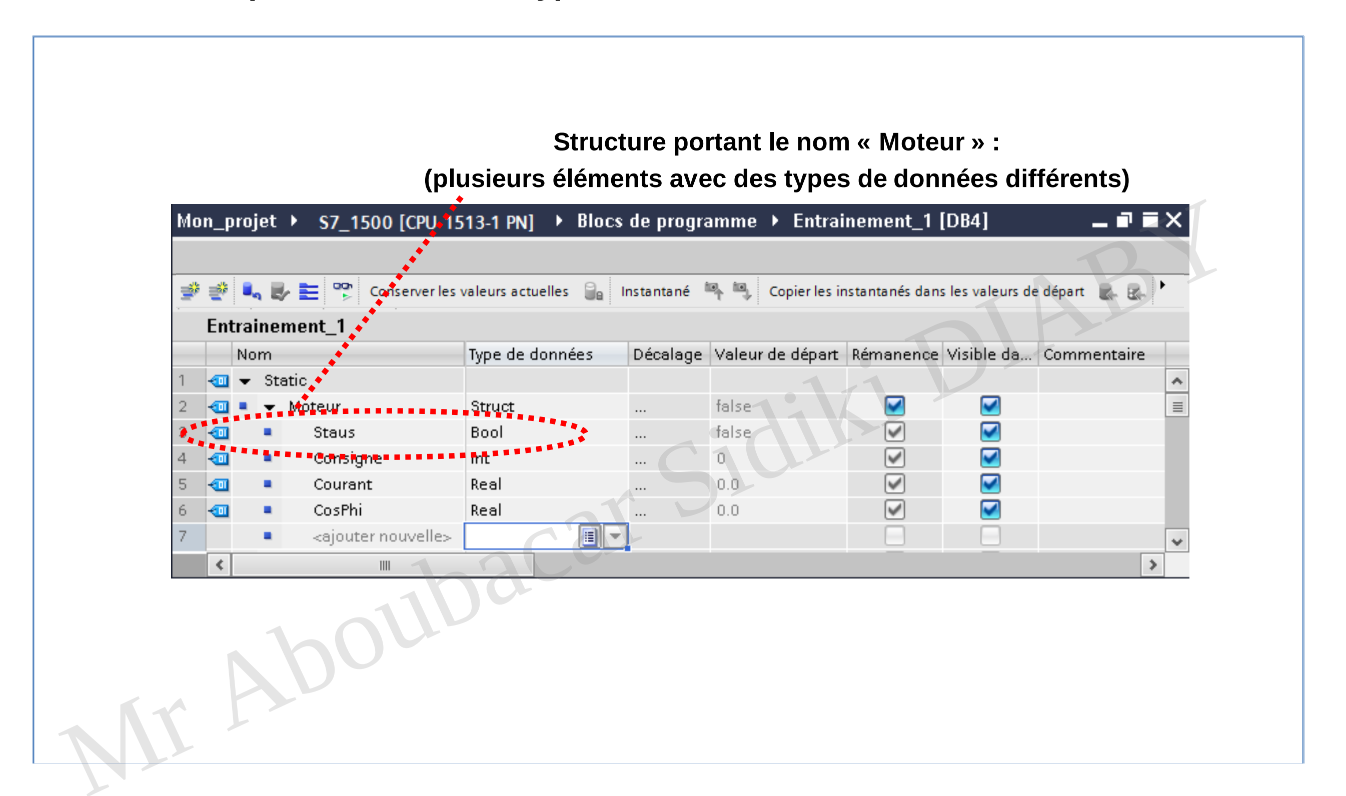
Tableau

Un tableau comporte plusieurs éléments qui sont définis avec le même type de données. L'illustration ci-dessus représente un tableau « Point de mesure » avec 10 éléments du type de données LREAL. Différentes valeurs de mesure seront enregistrées dans ce tableau par la suite.

Définir un tableau dans le DB

Le type de données utilisé pour une variable de type tableau est « ARRAY[n..m] ». Le premier élément (n) et le dernier élément (m) sont indiqués entre crochets. Dans notre exemple, [1..10] signifie 10 éléments, le premier élément étant adressé avec l'indice [1] et le dernier avec l'indice [10]. Au lieu de [1..10], on peut aussi définir [0..9] par exemple, cela n'aura d'incidence que sur l'accès aux éléments.

11.5.2. Exemple de variable de type STRUCT



Structure

Cette image montre un exemple de structure nommée : “Moteur”. Un structure se compose de plusieurs éléments de type différents. Un élément individual d’une structure peut être de type élémentaire ou complexe.

L’accès à un élément de structure se fait par l’intermédiaire du nom de la stucture suivi du nom de l’élément, ce qui rend le programme plus facile à lire.

Exemple: accès à un élément individual dans une structure:

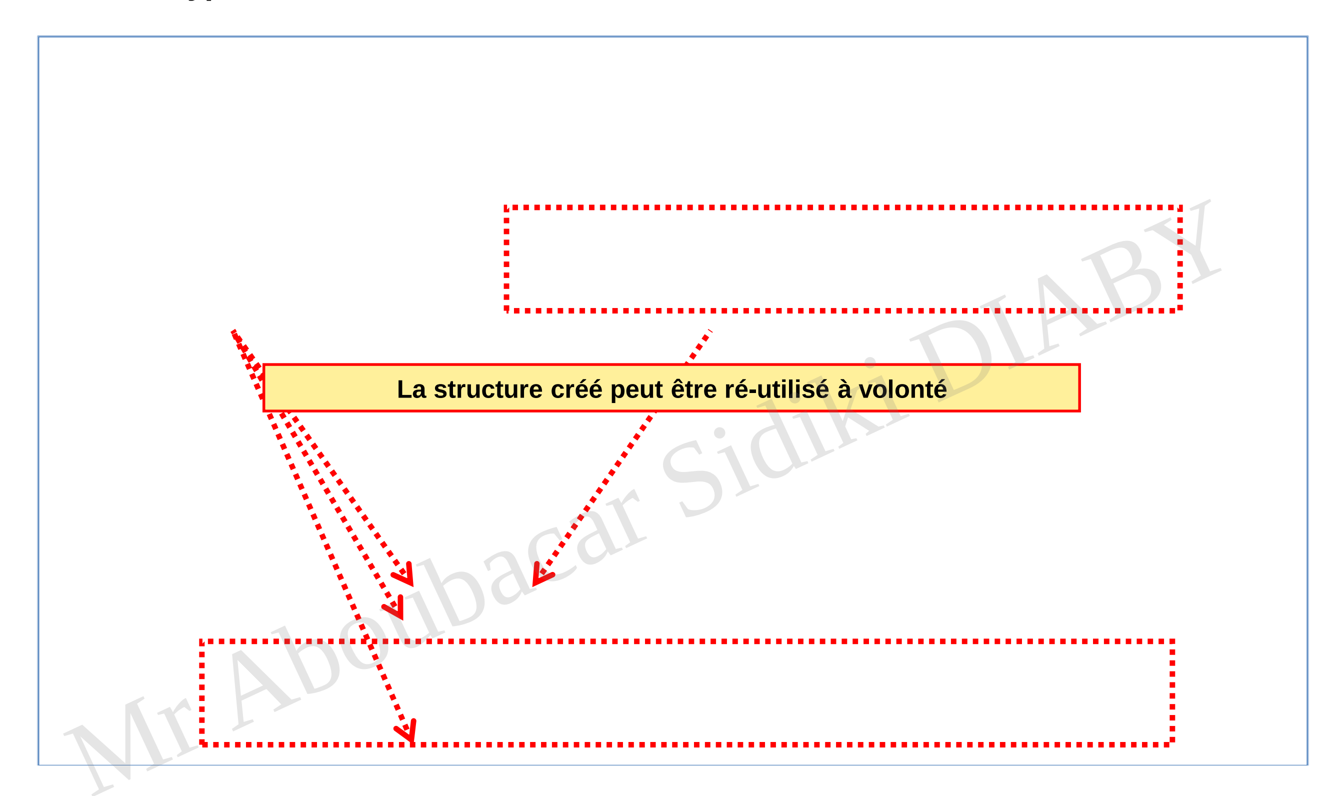
- "Entrainement_1".Moteur.Staus
- "Entrainement_1".Moteur.Courant

"Entrainement_1" est le nom symbolique du bloc de données qui contient la structure. Après le nom symbolique, (séparé par un point) le nom de la structure est spécifié. Se succeed ensuite le nom de l’élément (séparé par un point).

Déclaration d’une structure dans un DB

Une structure démarre toujours par le mot clé "STRUCT". L’identification de la fin de la structure est systématiquement "END_STRUCT". Un nom est renseigné dans cet intervalle pour définir la structure (dans l’exemple: "Moteur").

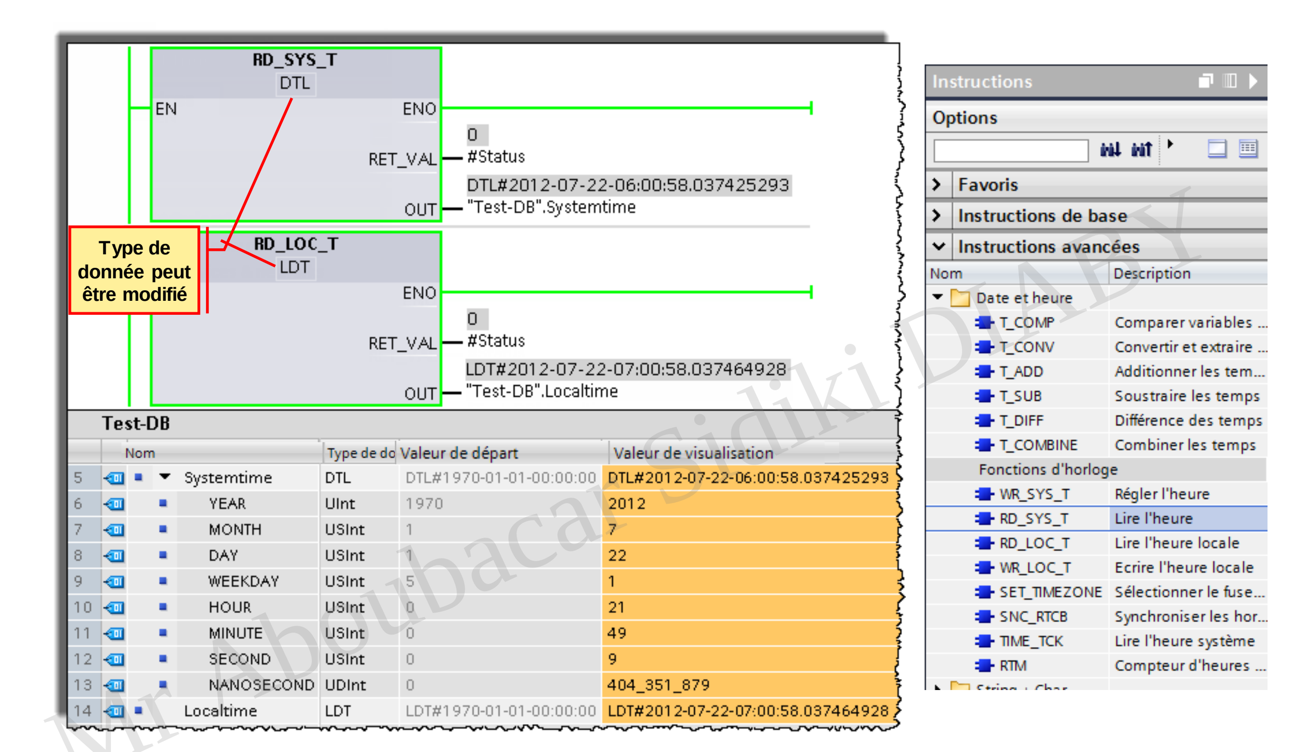
11.5.3. Type de donnée API



Types de données PLC

Les types de données API sont des types de données que vous avez définis et qui sont utilisés comme modèles pour déclarer les paramètres et les variables de types de données complexes (par exemple, les variables de structure). Les types de données de l'automate sont créés et stockés dans le dossier des types de données de l'automate et contiennent une structure de données constituée de types de données élémentaires et / ou complexes. Dans la déclaration d'une variable en fonction du type de données API, une variable de structure est créée dont la structure de données interne est définie par le type de données de l'automate. Les types de données API peuvent être utilisés pour la déclaration des variables dans les blocs de données globaux et dans les blocs pour la déclaration des variables locales (TEMP, STAT) ainsi que les paramètres (IN, OUT et INOUT).

11.5.4. Fonctions RD_SYS_T et RD_LOC_T



Dans les instructions de la fiche de tâches> Instructions avancées> Date et heure du jour, vous trouverez des fonctions et des instructions spécialement conçues pour le traitement des types de données Date / Heure.

Vous trouverez ici des fonctions avec lesquelles, par exemple, les temps peuvent être liés, comparés, écrits, lus, etc., les uns avec les autres.

RD_SYS_T

Vous utilisez cette instruction pour lire la date actuelle et l'heure actuelle (heure du module) de l'horloge du processeur.

RD_LOC_T

Vous utilisez cette instruction pour lire l'heure locale actuelle à partir de l'horloge du processeur et la sortir à la sortie OUT. Pour afficher l'heure locale, les informations sur le fuseau horaire ainsi que le début de l'heure d'été et de l'heure standard que vous avez définis lors de la configuration de l'horloge de la CPU sont utilisés.

Sommaire 12

12. Liaison HMI

A l'issue du chapitre, vous allez ...



... savoir paramétrer l'interface du pupitre opérateur à écran tactile

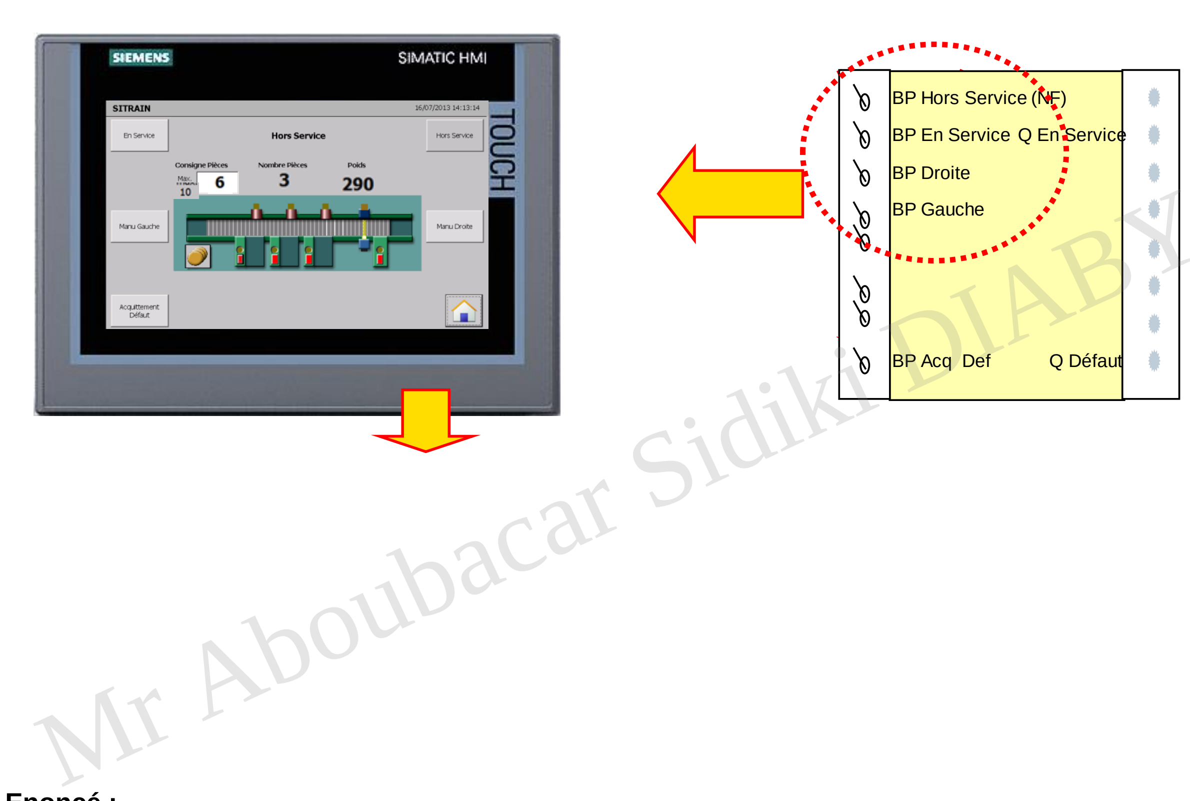
... savoir expliquer le principe de l'échange de données entre un pupitre opérateur à écran tactile et une CPU via des variables

... savoir mettre en service un projet avec un pupitre opérateur à écran tactile

… savoir adapter un programme STEP 7



12.1. Enoncé : Contrôler la machine via le panneau tactile



Enoncé :

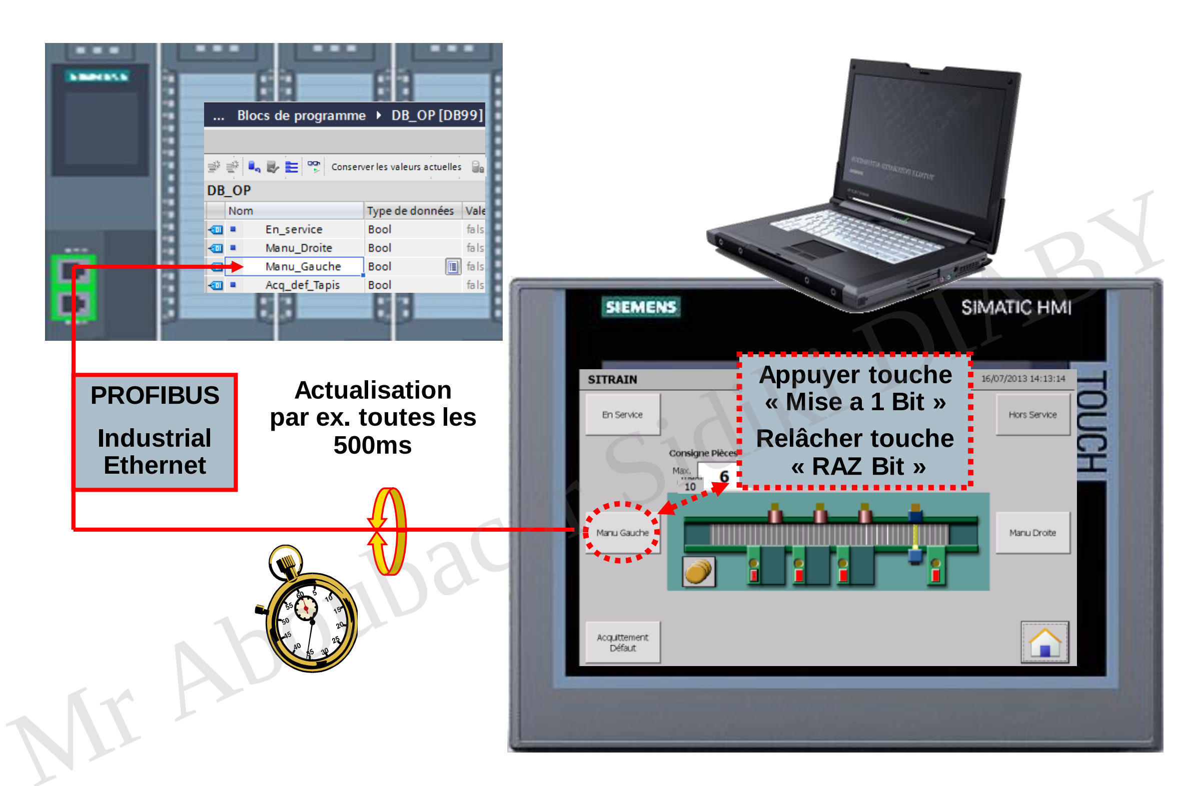
Le projet de panneau tactile doit être mis en service et le programme S7 du contrôleur doit être ajusté de manière à ce que ...

• les fonctions "Opération ON / OFF" et "Jog Right / Left" ne sont plus réalisées via les commutateurs du simulateur mais via les boutons correspondants sur l'écran tactile.

• L'acquittement d'un défaut de convoyeur doit toujours être possible via le commutateur de simulateur "S_Acknowledge", et, en plus, via le bouton correspondant sur l'écran tactile.

• La quantité SETPOINT n'est plus une constante 3, mais peut être prédéfinie via un champ de saisie sur l'écran tactile.

12.2. Echange de données entre le pupitre opérateur et la CPU



Echange de données entre le pupitre opérateur à écran tactile et la CPU

L'échange de données entre le SIMATIC S7 et le système IHM s'effectue à l'aide de variables. Il faut pour ce faire relier, lors de la configuration du terminal IHM, les objets graphiques, comme les touches de l'écran tactile et les zones d'entrée/sortie, avec les variables IHM, qui doivent à leur tour être reliées avec les variables API de la CPU. Le système IHM échange cycliquement des valeurs entre ces variables. Le transfert de données entre SIMATIC S7 et le système IHM s'opère de manière cyclique (les variables de process sont lues cycliquement par le terminal IHM en fonction du temps d'actualisation configuré).

Variables IHM

Les variables IHM peuvent être reliées avec les variables API globales ou avec les plages de données globales suivantes de la CPU :

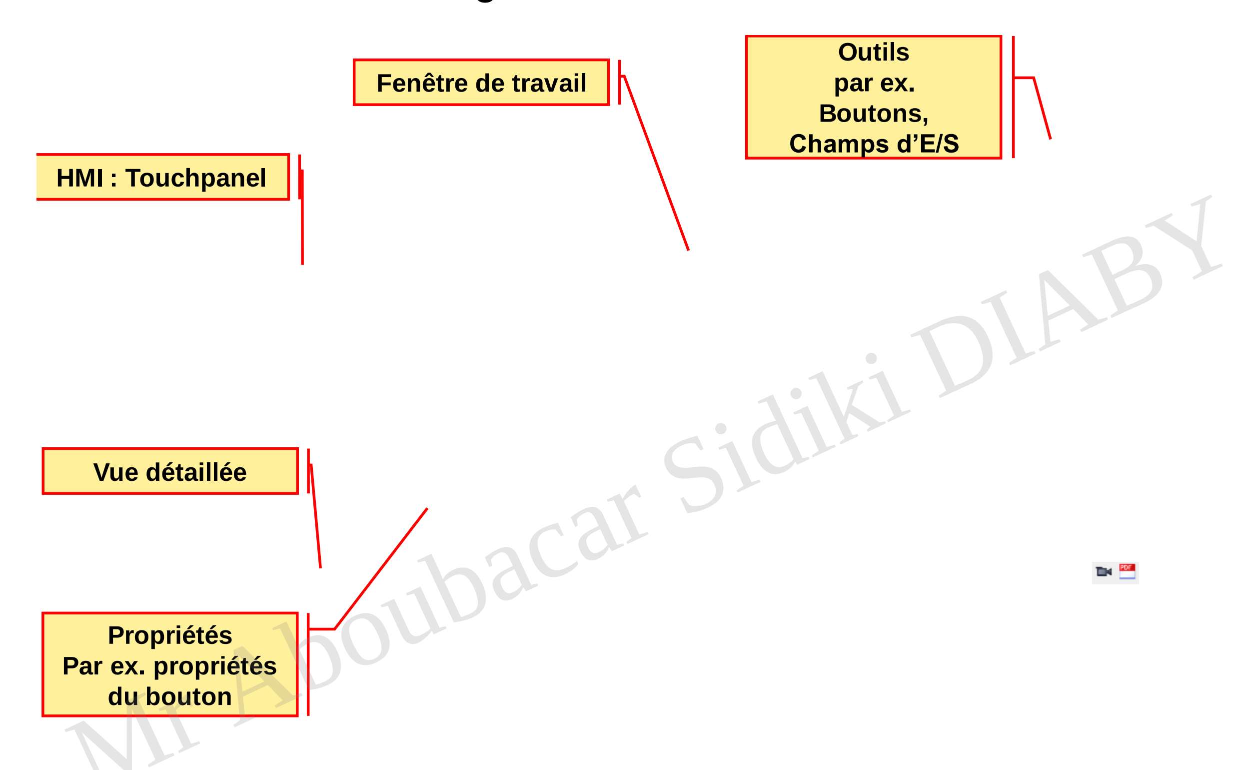
- Blocs de données (DB)
- Mémentos (M)
- Entrées (I) et sorties (Q)

Les systèmes IHM disposent par ailleurs de variables locales non reliées au processus (ces variables sont traitées exclusivement de manière interne et n'utilisent aucune ressource de communication).

Communication

Les pupitres opérateur peuvent communiquer avec l'automate via les systèmes de bus PROFIBUS ou Industrial Ethernet. La communication fait appel au protocole S7 et est organisée par les systèmes d'exploitation de la CPU S7 et du système IHM. L'utilisateur n'a aucune programmation à réaliser. Les pupitres opérateur peuvent également échanger des données avec plusieurs automates.

12.3. Interface de configuration WinCC



Fenêtre de projet

Le Navigateur du projet affiche l'arborescence de tous les appareils avec leur configuration et leur paramétrage. A partir de là, vous pouvez accéder aux différents éditeurs. Il donne aussi accès au paramétrage des langues du projet et à la gestion des versions.

Zone de travail

Il s'agit de la zone de configuration centrale dans laquelle les objets du pupitre opérateur sont traités avec l'éditeur actif. Il est possible d'ouvrir simultanément plusieurs éditeurs.

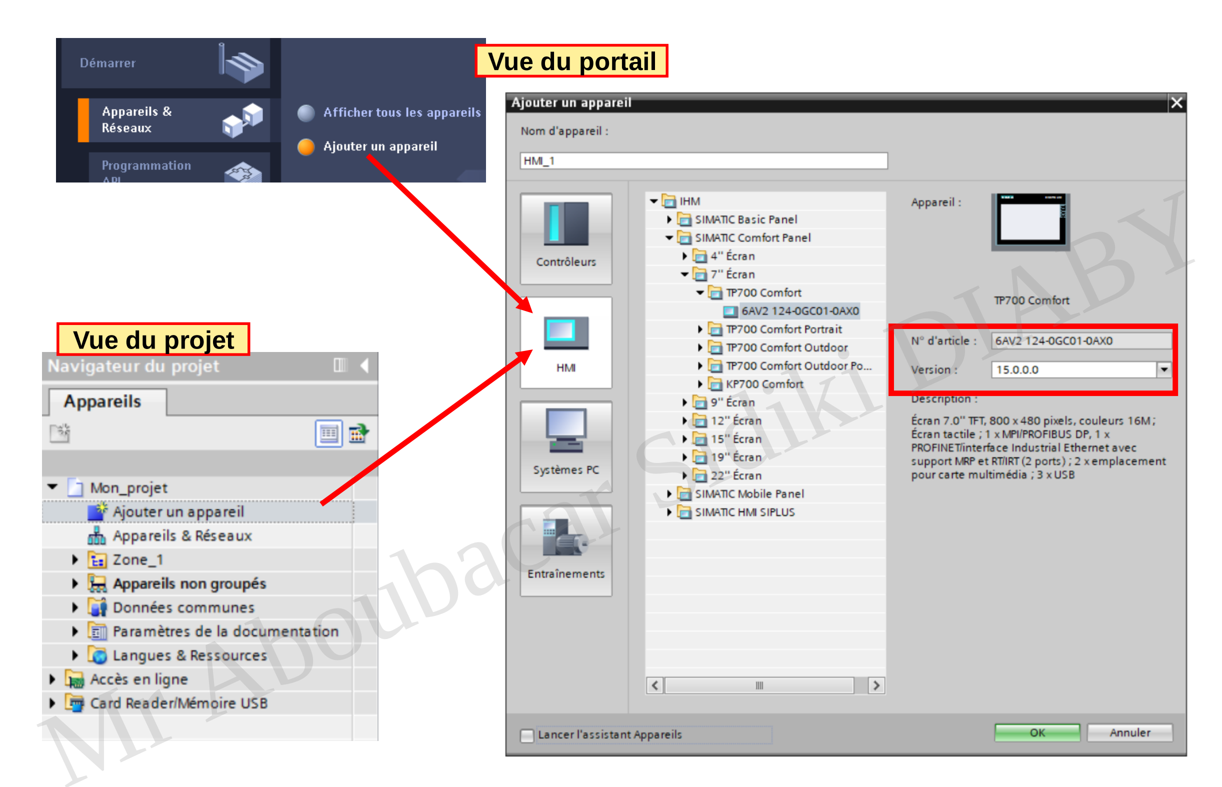
Fenêtre des propriétés

Les propriétés des objets sélectionnés (par ex. des vues, objets graphiques, variables) peuvent être éditées dans la fenêtre « Propriétés ». Cette fenêtre est uniquement disponible pour les éditeurs permettant de définir des propriétés d'objet.

Fenêtre d'outils

La fenêtre d'outils contient tous les objets qui peuvent être configurés dans les vues et donne l'accès aux bibliothèques.

12.4. Ajouter un appareil HMI

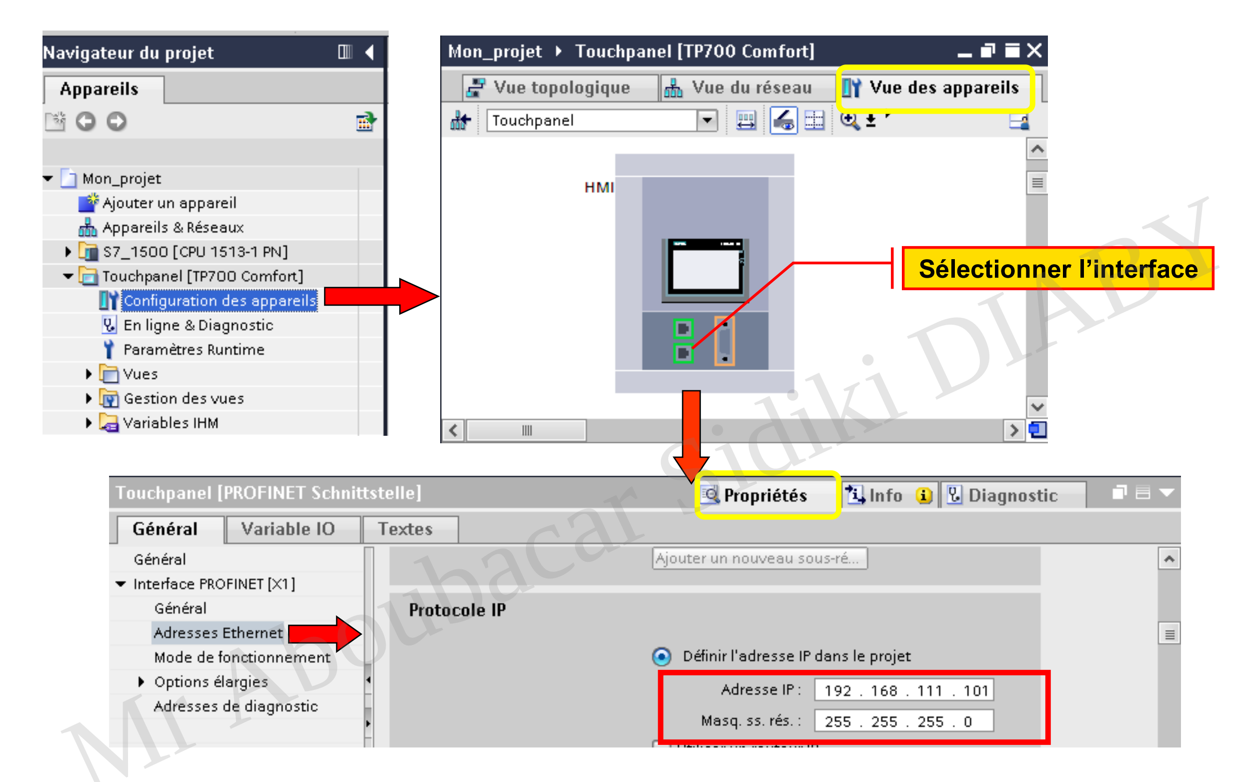


Ajouter un appareil HMI

Les nouveaux appareils HMI peuvent être ajoutés aussi bien à partir de la vue du portail que de la vue du projet.

Il faut veiller à saisir les bonnes caractéristiques de l'appareil, comme le numéro de référence et le numéro de version.

12.4.1. Paramétrer l’adresse IP du pupitre opérateur



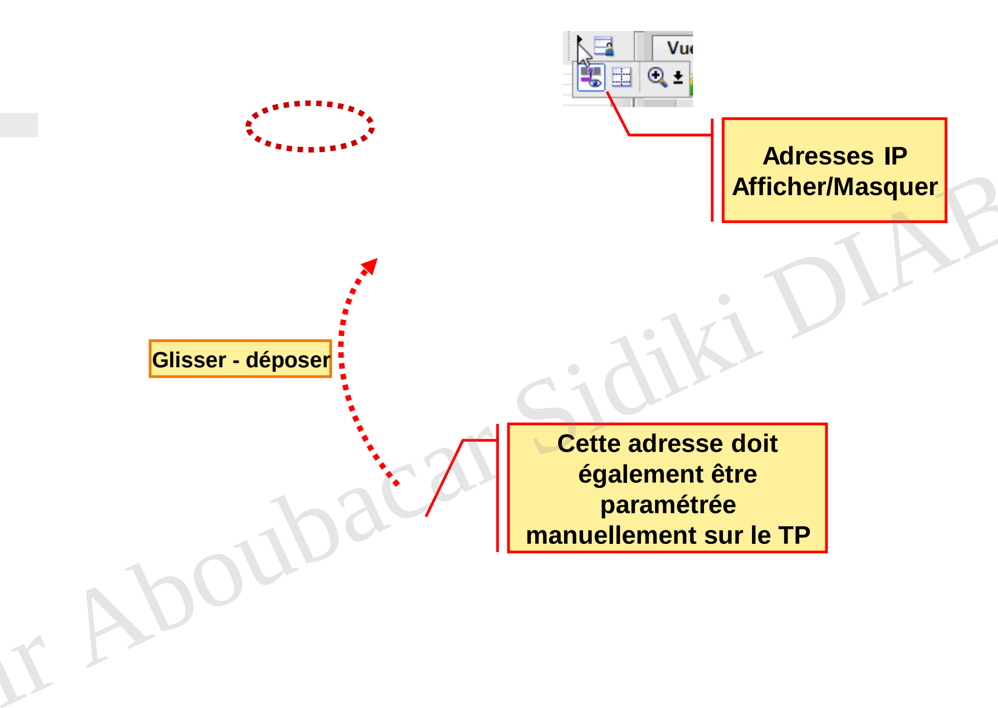
Interface PROFINET du pupitre opérateur

L’adresse IP de l’appareil est paramétrée dans l’éditeur « Appareils & Réseaux », dans la « Vue des appareils ». Il est plus simple d’appeler directement la vue à partir de la « Configuration des appareils » dans le navigateur du projet.

Le paramétrage de l’interface PROFINET (adresse IP et masque de sous-réseau) s’effectue dans la fenêtre d’inspection, sous l’onglet « Propriétés ».

Si une liaison doit être établie entre l’appareil HMI et la CPU, les deux appareils doivent se voir affecter le même masque de sous-réseau et des adresses IP situées dans le même sous-réseau.

12.4.2. Mettre le pupitre opérateur en réseau



Mettre le pupitre opérateur en réseau

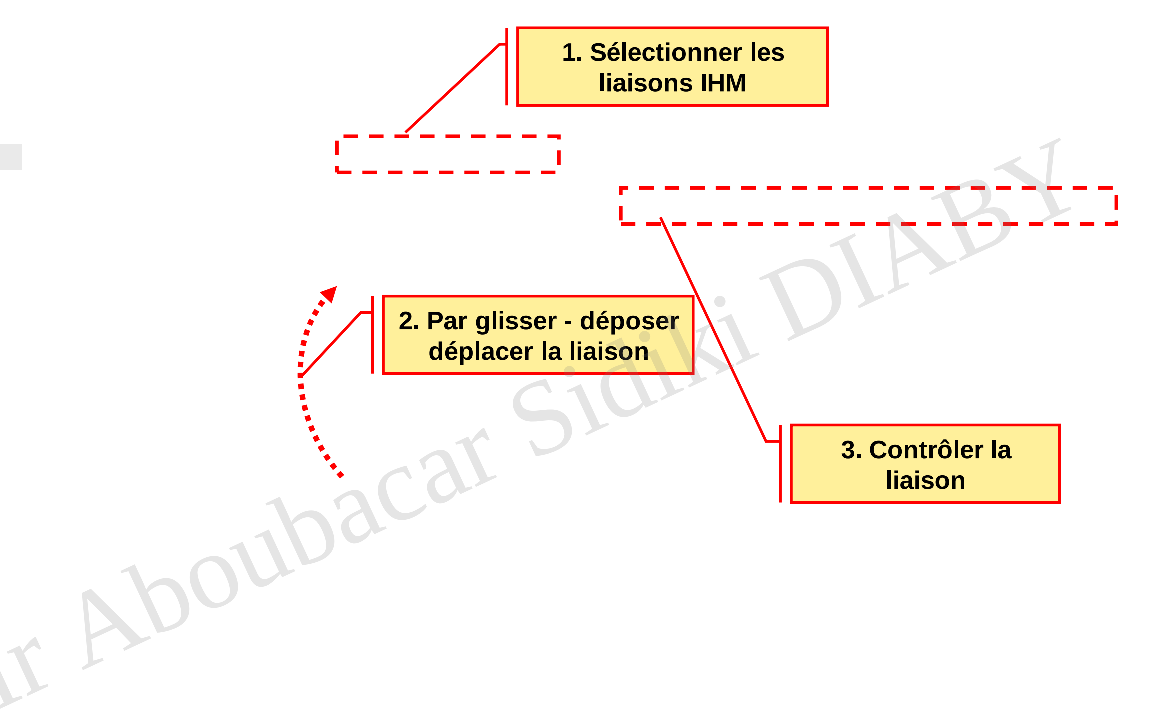
La mise en réseau des appareils s'effectue avec l'éditeur « Appareils & Réseaux » dans la « Vue du réseau », onglet « Mise en réseau ».

Les interfaces des appareils sont reliées entre elles par glisser-déposer.

☞ Lors de la mise en réseau, seuls sont reliés les appareils avec un sous-réseau. L'interface des appareils doit également être compatible avec le type de réseau.

Il est donc important de paramétrer au préalable l'adresse IP et le masque de sous-réseau de l'appareil HMI.

12.4.3. Paramétrer la liaison HMI



Paramétrer la liaison HMI

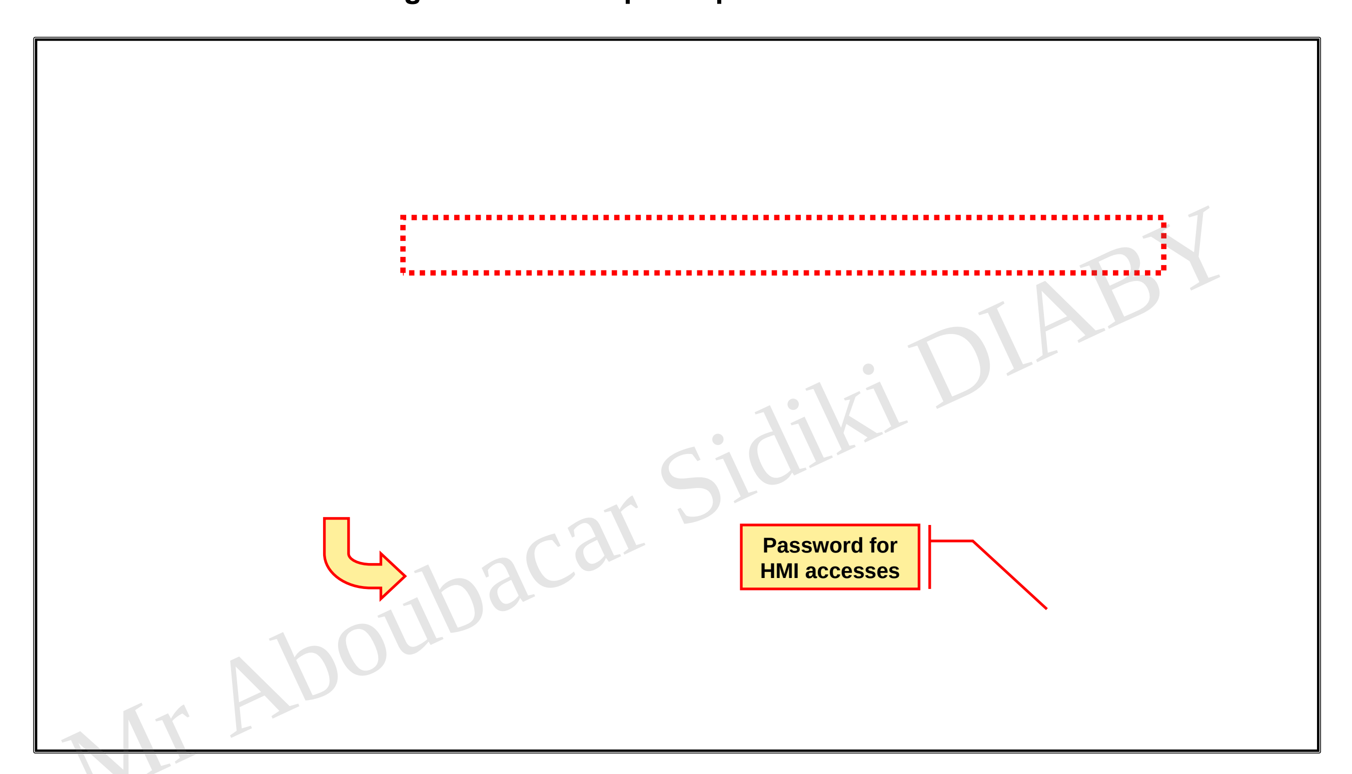
Lors de la configuration d'une ou plusieurs liaisons HMI, on définit les partenaires de communication avec lesquels l'appareil HMI devra échanger des données en cours d'exploitation.

L'appareil HMI peut également être relié ou échanger des données avec plusieurs automates.

Le réseau peut aussi contenir des automates avec lesquels l'appareil HMI n'échange aucune donnée. L'appareil HMI est alors « mis en réseau » avec ces automates, mais non « relié ».

 Si l'on configure une liaison HMI sans avoir préalablement relié l'appareil HMI avec l'automate, l'appareil HMI est mis en réseau en arrière-plan avec la CPU.

12.4.3.1. Vérifier et renseigner un mot de passe pour l'accès à une CPU



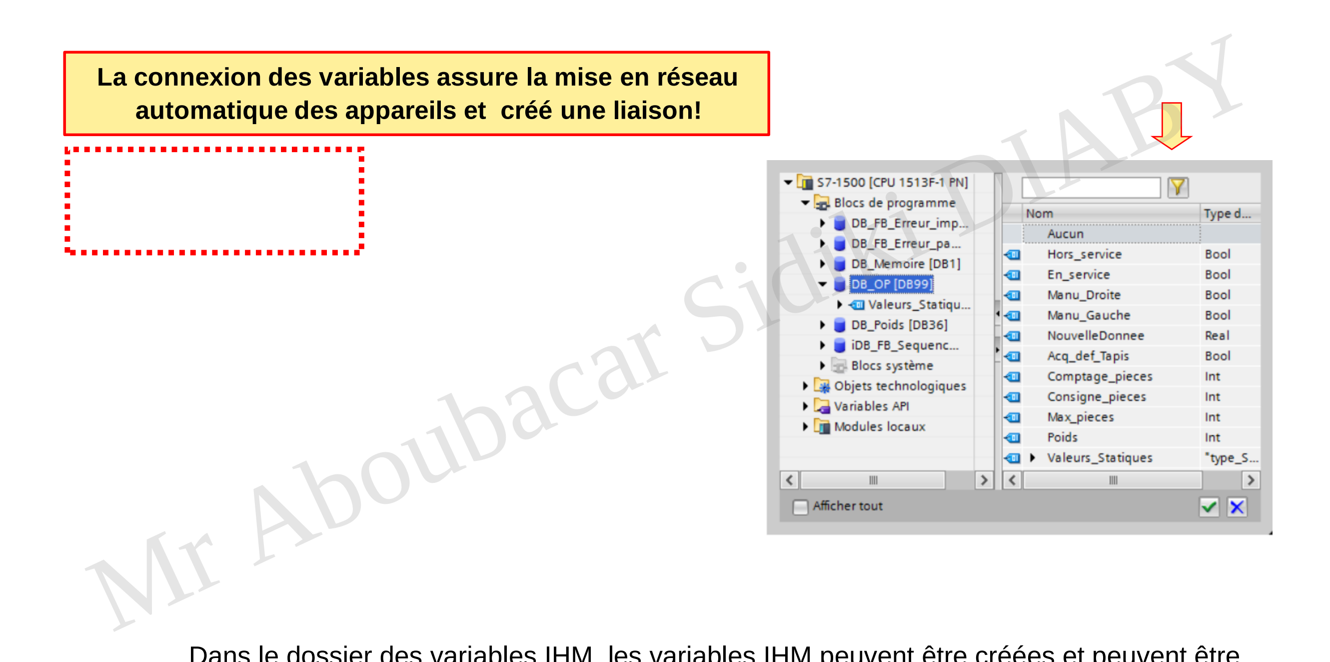
Connection:

Les connexions configurées sont visibles dans les Connexions du pupitre opérateur.

Requête de mot de passe pour les accès IHM

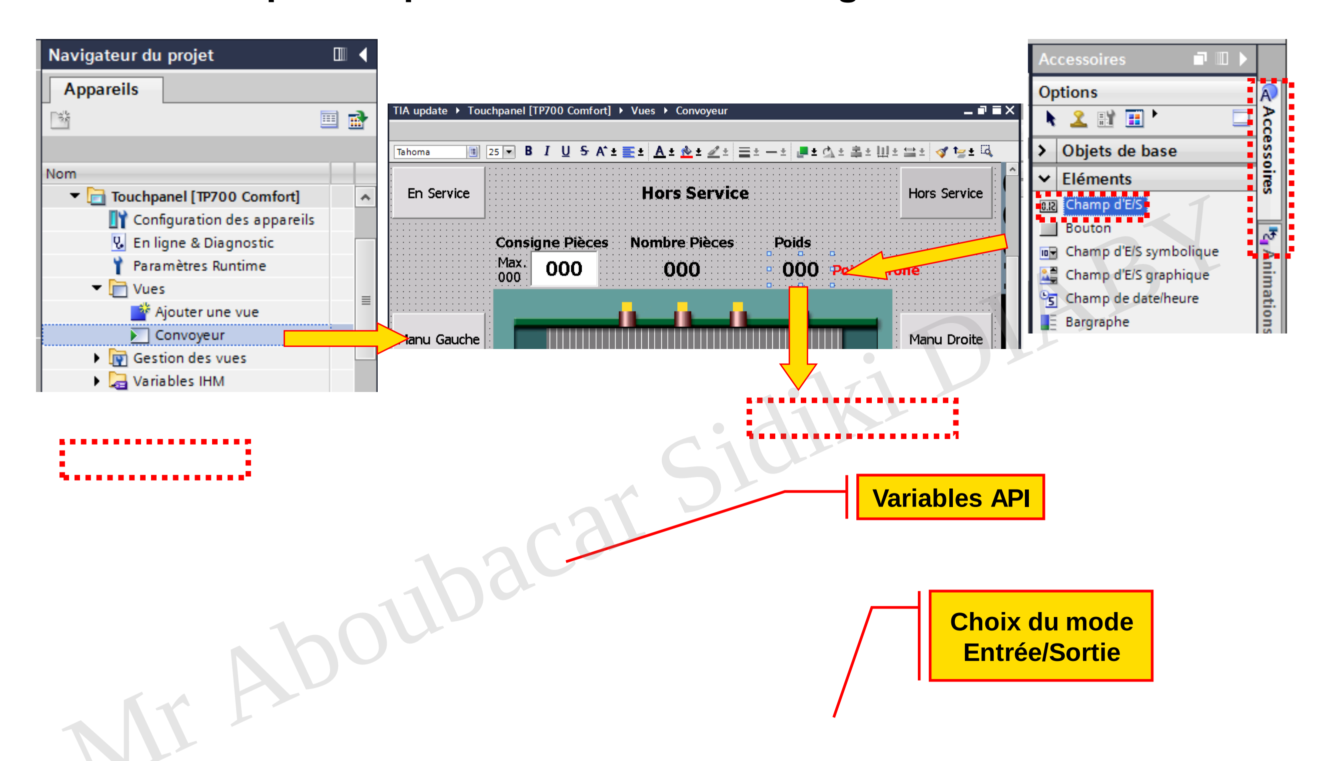
Afin d'obtenir les droits d'accès à une CPU protégée par mot de passe, le pupitre opérateur doit se connecter à la CPU avec un mot de passe au démarrage de Runtime. Ce mot de passe doit être spécifié dans la configuration Connections du pupitre opérateur (voir image en bas à droite).

12.4.4. Création de variables IHM et connexion avec des variables automate.



Dans le dossier des variables IHM, les variables IHM peuvent être créées et peuvent être connectées avec les variables API associées via une connexion existante. Les propriétés, telles que le mode d'accès, le cycle d'acquisition, etc. peuvent être définies.

12.5. Champ d'E/S pour la saisie et l'affichage des valeurs



Champ d'E/S :

Les champs en mode « Sortie » affichent les valeurs des variables, les champs d'entrée permettent la saisie des valeurs des variables. Le mode est réglable dans la fenêtre « Propriétés ».

Mode :

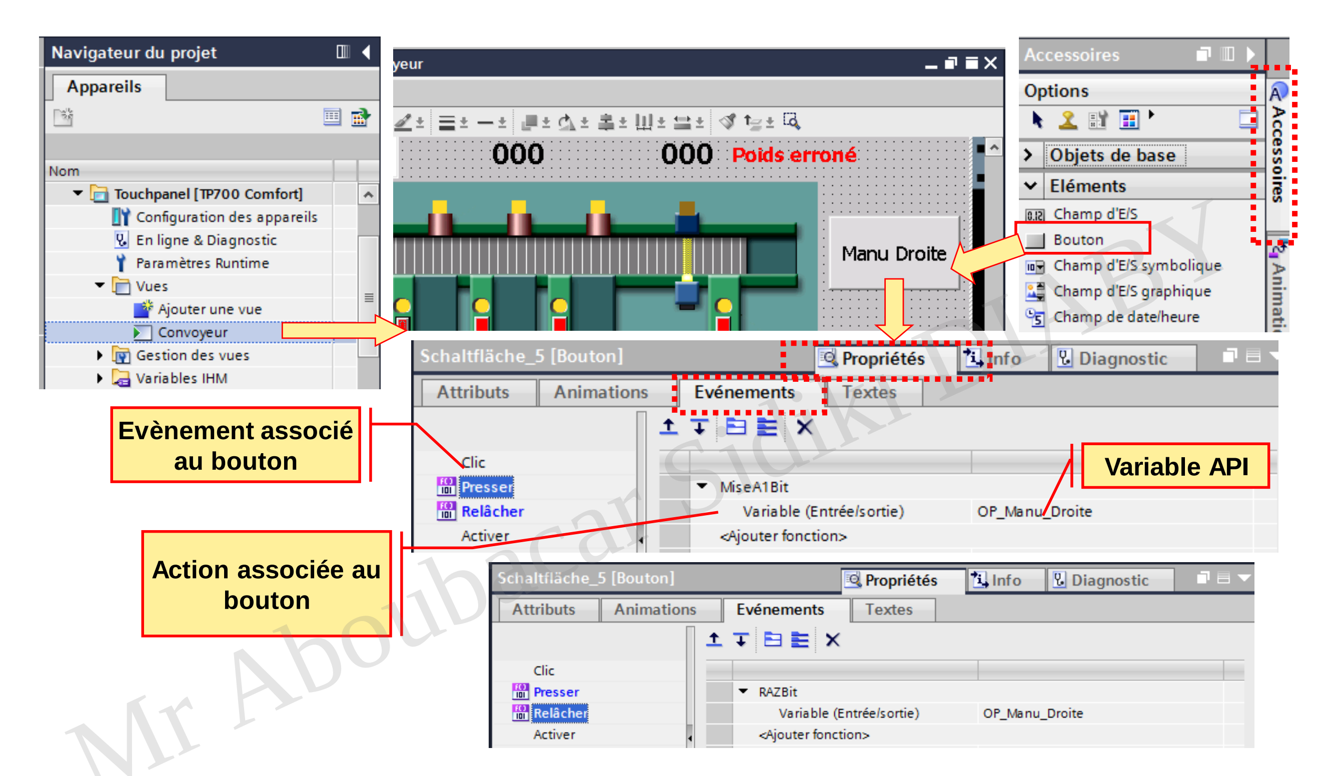
- Sortie
  La valeur de la variable est uniquement visualisée. La variable est lue et actualisée en fonction du cycle d'acquisition paramétré.
  Une modification de valeur (saisie) n'est pas possible à partir du pupitre opérateur.

- Entrée/Sortie
  La valeur de la variable est affichée. La variable est lue et actualisée en fonction du cycle d'acquisition paramétré.
  Une modification de la valeur (saisie) à partir du pupitre est possible. Pour les pupitres tactiles la modification s'effectue via le clavier virtuel qui s'affiche automatiquement, pour les pupitres à touches elle s'effectue via le clavier de l'appareil.

Commentaire :

Un champs d'entrée/sortie peut être directement créé en faisant cliquer/glisser une variable automate depuis une table ou un bloc de données (ou avec l'aide de la vue détail). Par ce procédé, une variable IHM est automatiquement créé et connecté sur la liaison existante entre IHM et automate.

12.5.1. Bouton pour exécution de la fonction



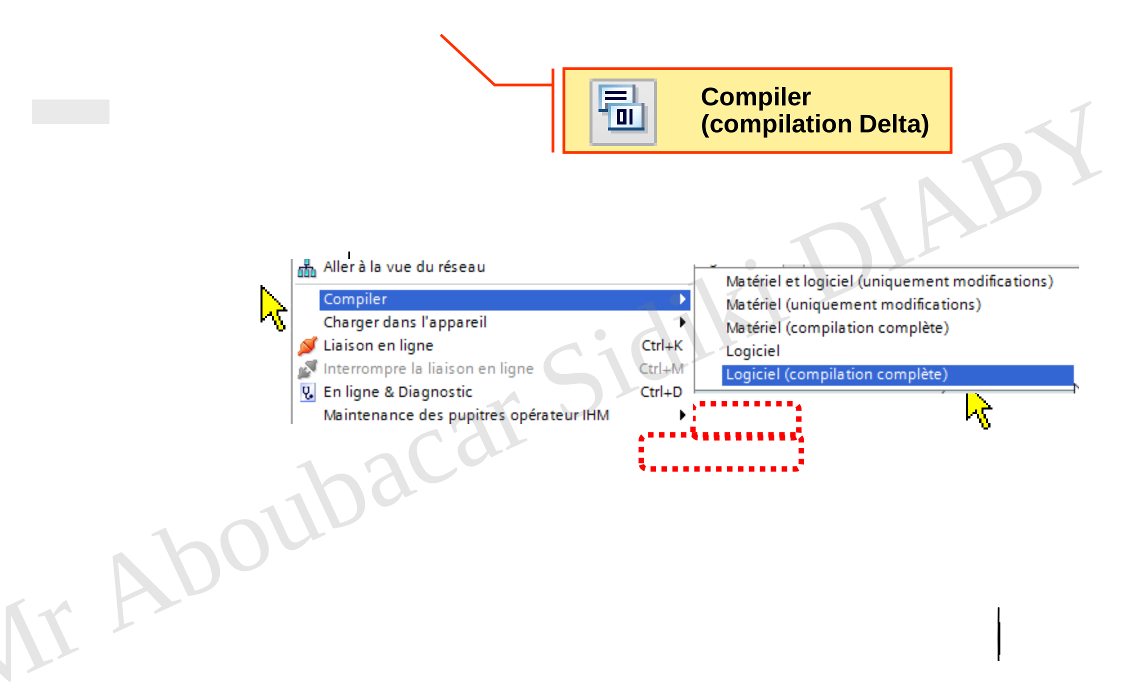
Bouton

Les boutons servent à commander l'exécution d'une ou plusieurs fonctions associées à un événement de commande.

Ils permettent à l'opérateur de déclencher des fonctions système, comme la sélection d'une vue ou la mise à « 1 » ou à « 0 » d'une variable affichée dans la vue.

« Evénements » permet de définir l'événement qui doit déclencher l'exécution d'une ou plusieurs fonctions système et de définir ces fonctions système.

12.5.2. Compilation complète du projet



Compiler (changement seulement) 

Toutes les modifications depuis la dernière compilation sont recompilées.
→ Compilation Delta

Logiciel/Matériel (compilation complète)

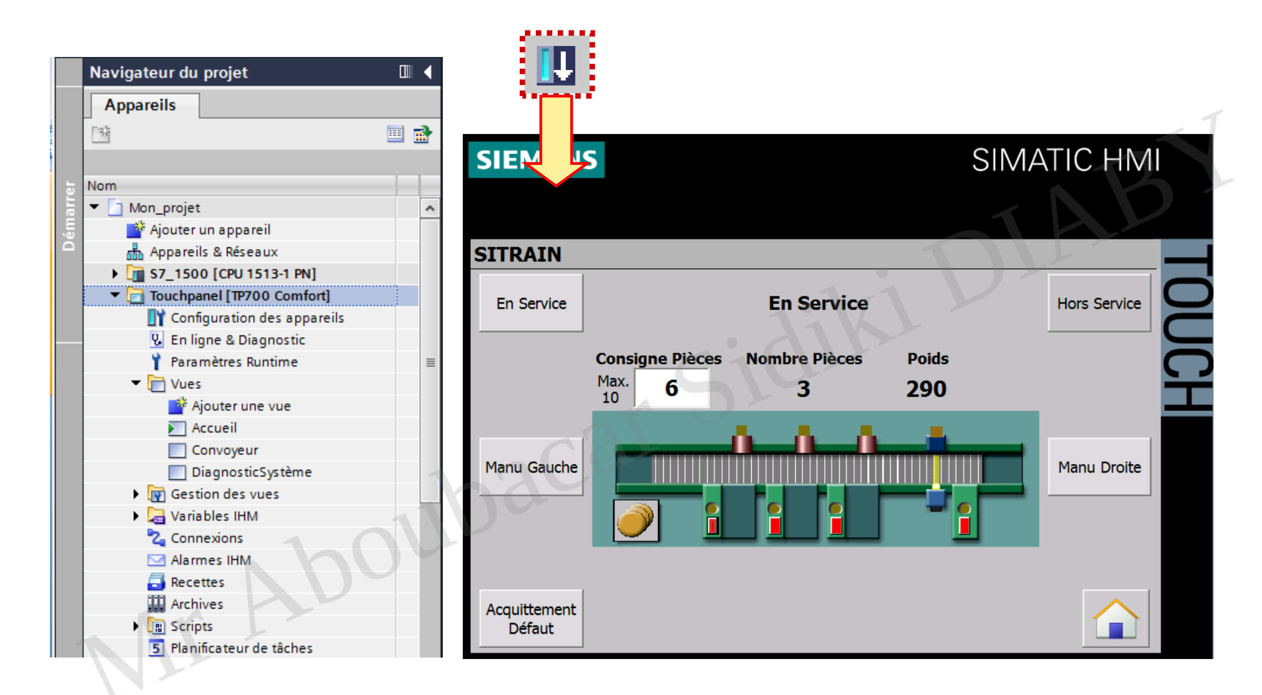
Il s'agit d'une nouvelle compilation complète du pupitre.

Quand est ce nécessaire ?

Lorsque des fonctions correctement configurées ne sont pas correctement exécutées ou même pas du tout exécutées.

Lorsque des messages d'erreurs « incohérents » apparaissent lors de la compilation.

12.6. Chargement du projet dans le HMI



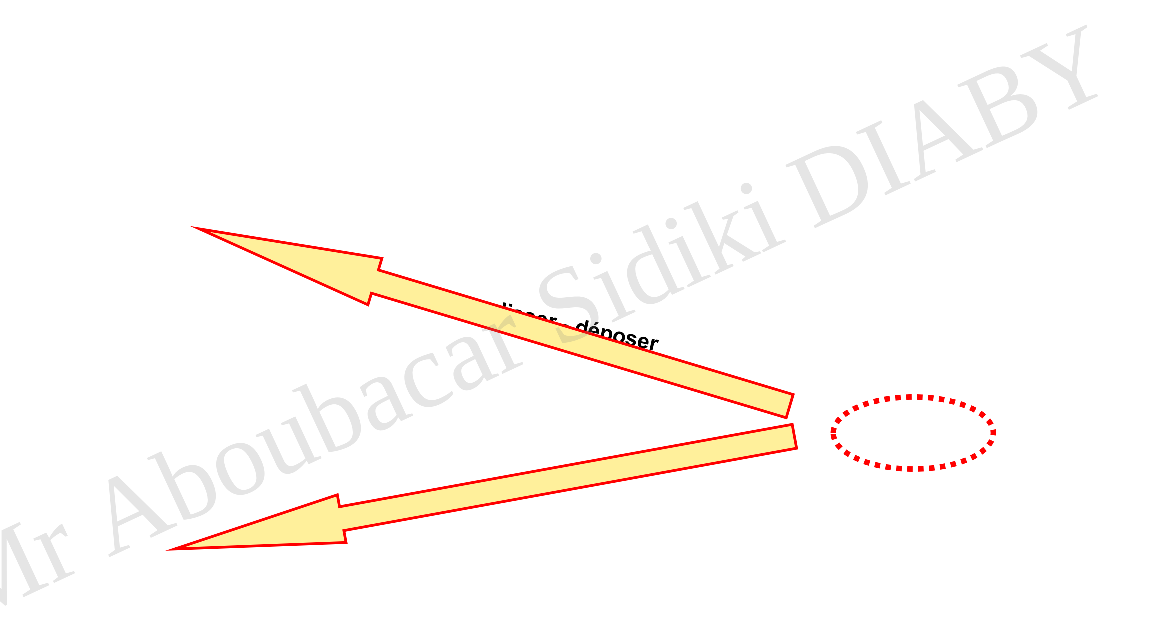
Charger le projet HMI dans le Touchpanel

Lors du chargement d'un projet HMI sur un ou plusieurs pupitres opérateur, la partie modifiée du projet depuis le dernier transfert est automatiquement compilée avant d'être chargée. On est ainsi assuré de toujours transférer la dernière version du projet. Avant le début du chargement, on a par ailleurs, la possibilité d'activer l'option « Ecraser tous ».

Lors d'une mise en route, il est recommandé de compiler entièrement le projet via la commande « Compiler > Logiciel (compilation complète) » du menu contextuel du pupitre opérateur. Si des variables HMI associées à des variables API sont également utilisées dans le projet, tous les blocs STEP7 modifiés doivent également être compilés via la commande : « Compiler > Logiciel » du menu contextuel et chargés dans la CPU.

Pour réduire les temps de génération Delta pendant la configuration, il est également recommandé d'utiliser occasionnellement la commande : « Compiler > Logiciel (Compilation complète) ».

12.7. Exercice 1: copier le projet Touchpanel et le bloc de données DB_OP

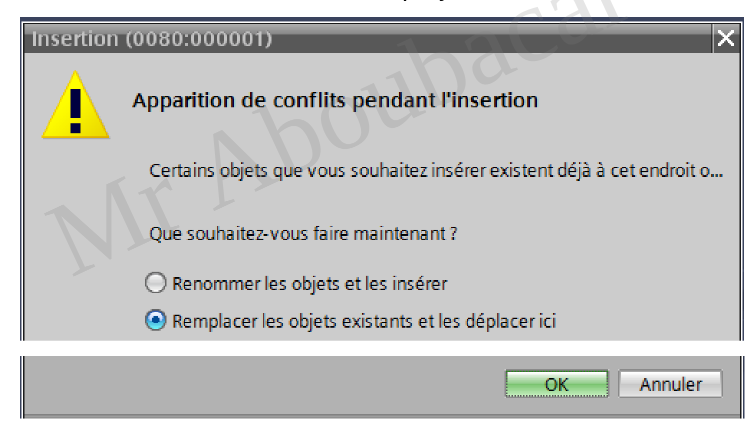


Enoncé

Votre projet ne contient pas encore d'appareil HMI. Au lieu de réaliser une configuration entièrement nouvelle, vous allez copier le projet « Touchpanel , à partir de la bibliothèque globale « Bib_TIA_PRO1 » dans votre projet.

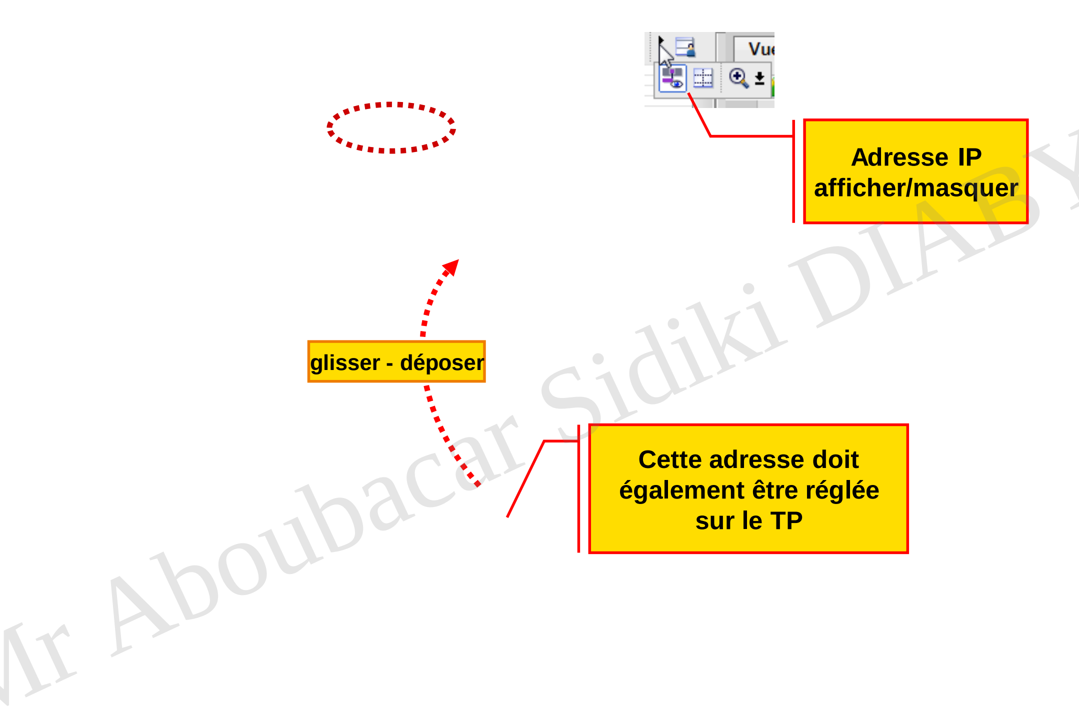
Marche à suivre

1. Ouvrez la bibliothèque globale « Bib_TIA_PRO1 ».
2. Copier par glisser – déposer l'élément de la bibliothèque "DB_OP" (DB99) dans votre projet (figure ci dessus) et confirmez dans le dialogue que vous vous voulez remplacer les objets existants dans le projet.



3. Copiez par glisser – déposer l'élément de la bibliothèque « Touchpanel » dans votre projet (voir ci dessus).
4. Enregistrez votre projet.

12.7.1. Exercice 2: Mettre le Touchpanel en réseau

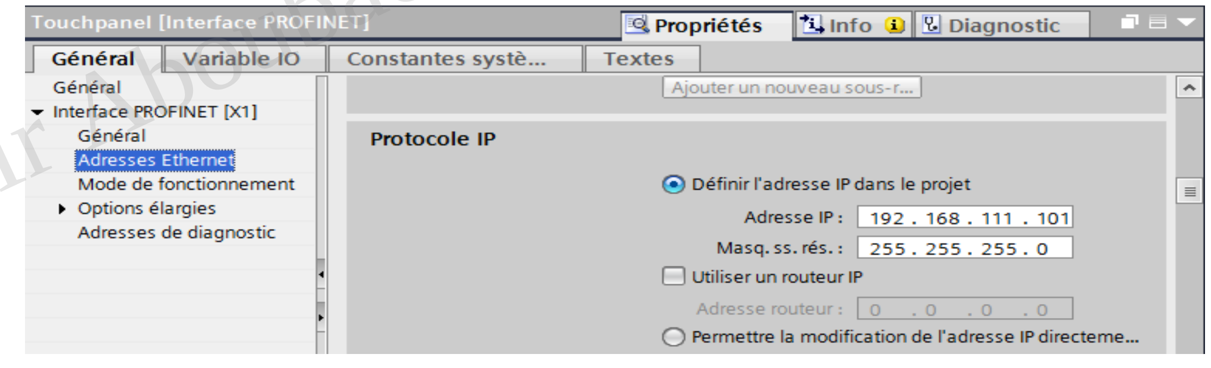


Enoncé

Le Touchpanel ajouté doit être intégré au réseau Ethernet.

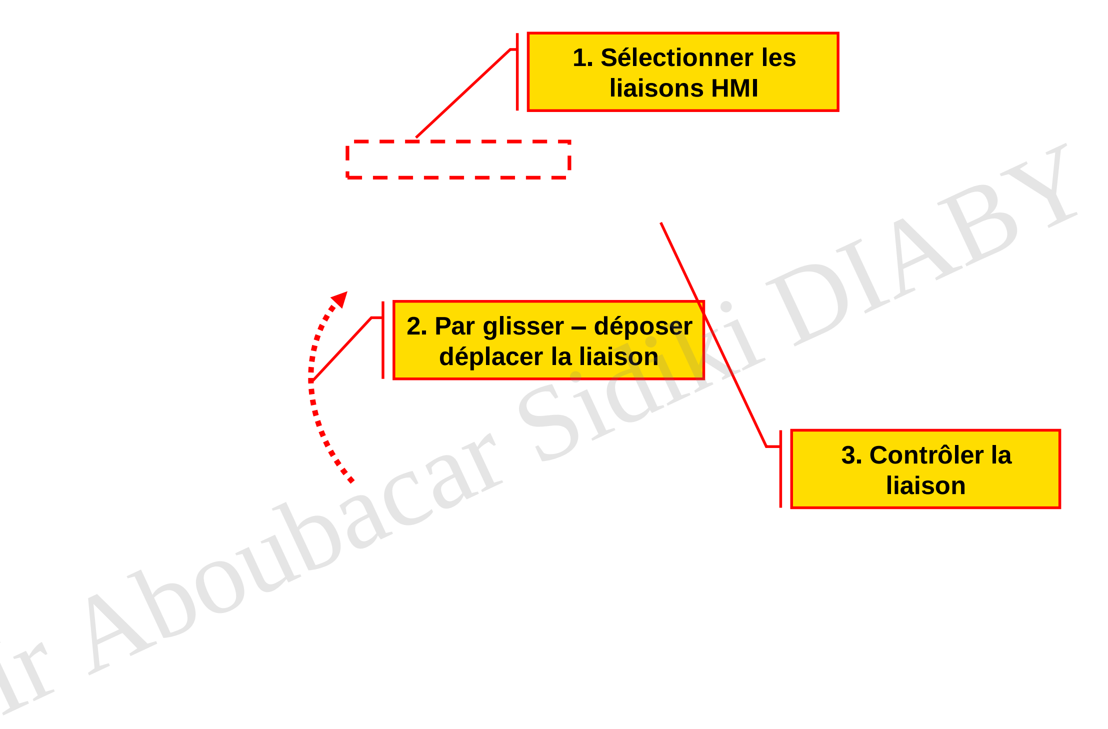
Marche à suivre

1. Démarrez l'éditeur « Appareils & Réseaux » dans le Navigateur du projet, basculez dans la « Vue du réseau » et retenez « Mise en réseau ».
2. Positionnez le pointeur de souris sur le petit carré vert de l'appareil HMI et tirez une liaison vers la CPU en maintenant le bouton gauche de la souris enfoncé. La mise en réseau sera réalisée, le sous réseau attribué et les paramètres relatifs à la mise en réseau (adresse IP et masque de sous réseau) seront attribués automatiquement.
3. Visualisez les adresses IP des appareils à l'aide du bouton détaillé dans la vue ci dessus.
4. Sélectionnez le Touchpanel et basculez sur la « Vue des appareils ».
5. Vérifiez et corrigez l'adresse Ethernet du Touchpanel dans la fenêtre d'inspection sous « Propriétés »:



6. Dans la vue Réseau, les interfaces de la CPU et de l'écran tactile doivent être connectées avec le même sous-réseau.

12.7.2. Exercice 3: Paramétrer la liaison HMI



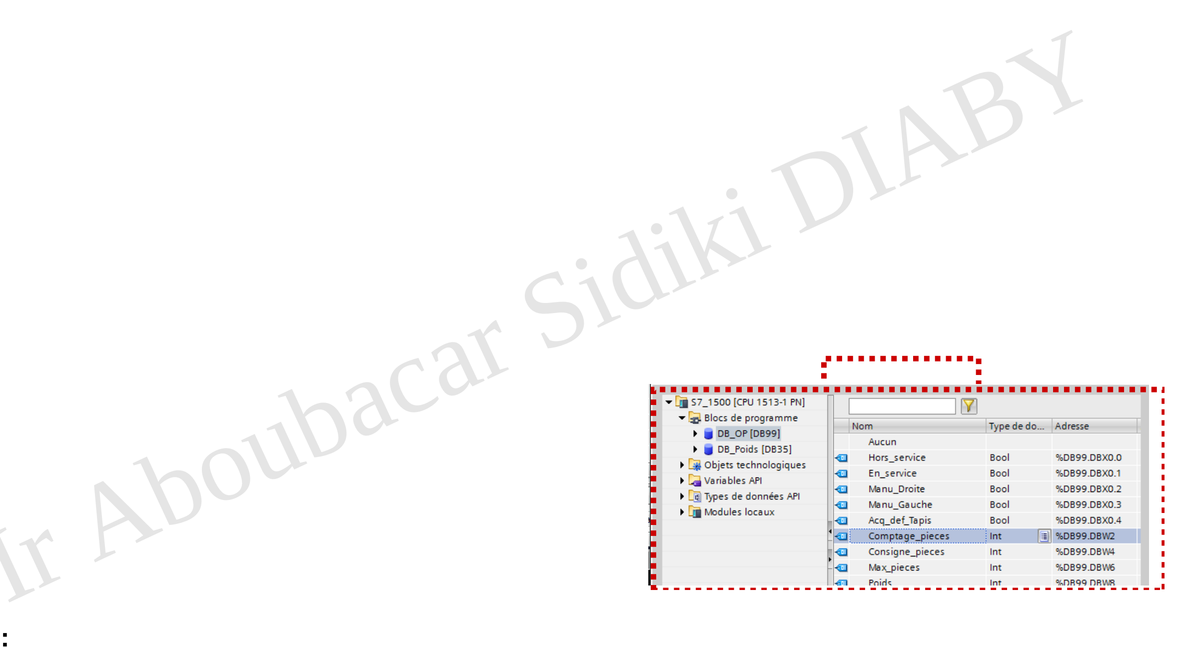
Enoncé

Après la mise en réseau Ethernet du TP, vous devez créer une liaison IHM entre le Touchpanel et la CPU. Le nom automatiquement attribué à la liaison « HMI_Liaison » peut être repris.

Marche à suivre

1. Lancez l'éditeur « Appareils et Réseaux dans le Navigateur du projet, basculez dans la « Vue du réseau » et choisissez « Liaison IHM » (voir figure).
2. Positionnez le pointeur de souris sur le petit carré vert de l'appareil HMI et tirez une liaison vers la CPU en maintenant le bouton gauche de la souris enfoncé. La liaison est établie et dotée du nom par défaut HMI_Liaison_1.
3. Vérifiez dans la fenêtre détaillée sous « Liaisons » si la liaison IHM a été créée correctement (voir figure).
4. Pour visualiser la nature de la liaison entre la CPU S7 et l'IHM, positionnez le pointeur de la souris sur la mise en réseau et choisissez « Liaison IHM » dans la fenêtre de dialogue.
5. Enregistrer votre projet

12.7.3. Exercice 4: Mettre à jour et Compléter les liaisons des variables HMI



Enoncé :

Les variables IHM "Q_En_Service », « M_DéfautTapis » et « Comptage_pièces » ne sont pas encore reliées à des variables API, pour d'autres variables, la « liaison » est éventuellement marquée en rouge. Vôtre tâche consiste maintenant à compléter les liaisons des variables manquantes et à corriger celles qui sont erronées.

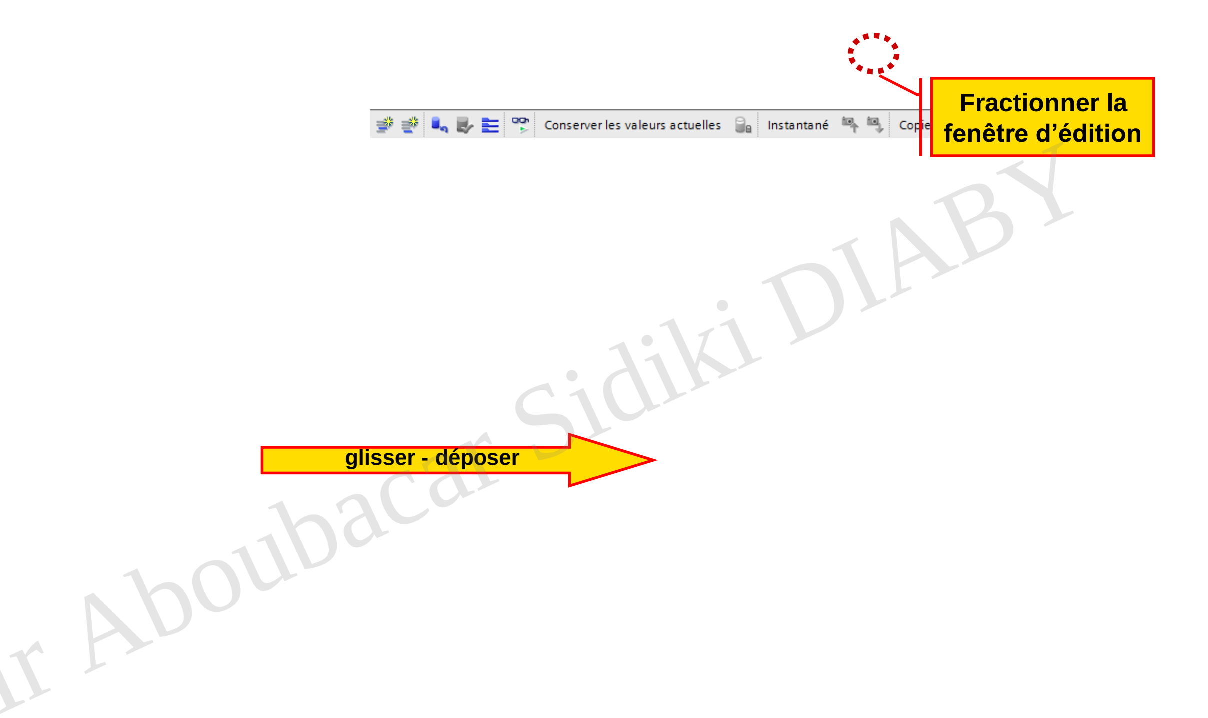
Marche à suivre :

1. Ouvrez « Variables IHM » de l'appareil IHM.
2. Corrigez les liaisons erronées de manière à faire disparaître toutes les entrées rouges dans la table des variables.
3. Adoptez la connexion des variables corrigées pour tous les autres tags. Pour ce faire, cliquez sur le coin inférieur droit de la cellule et, tout en maintenant le bouton gauche de la souris enfoncé, faites-le glisser sur toutes les autres cellules (fonction Excel familière). Confirmez la boîte de dialogue "Autocompletion" qui suit avec la sélection "Ecraser les attributs des variables".
4. Reliez les variables IHM qui ne sont pas encore reliées aux variables API correspondantes. Procédez comme indiqué dans la figure ce dessus.
5. Enregistrez le projet.

Remarque :

Pour connecter les variables, il faut choisir la bonne liaison pour toutes les variables IHM puis faire la reconnexion de celles-ci avec l'icône 

12.7.4. Exercice 5: Affichage de la consigne sur l‘afficheur



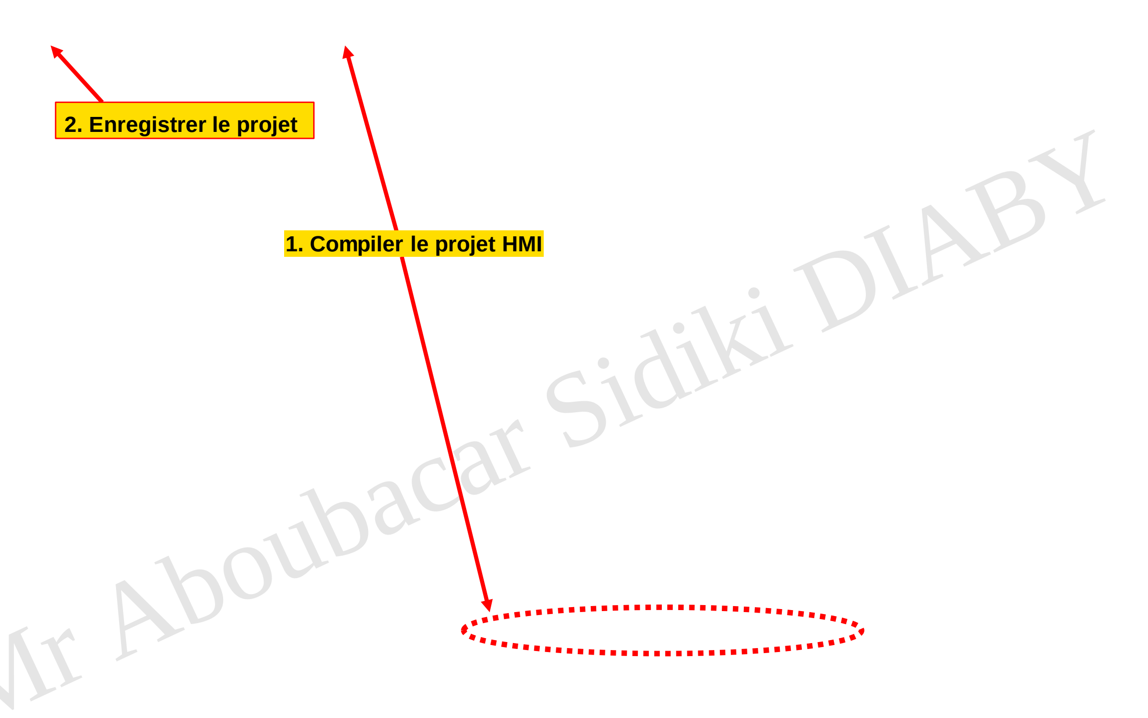
Enoncé

Jusqu'à présent, la consigne du nombre de pièces à transporter était fixée via la variable « DB_OP.Consigne_pieces » à la valeur 3. Elle doit désormais être saisie via un champ d'entrée sur le Touchpanel. Pour cela, il faut configurer un champ d'entrée/sortie supplémentaire dans la vue du Touchpanel (voir figure)

Marche à suivre

1. Fermez tous les objets ouverts dans l'éditeur.
2. Ouvrez le bloc de données « DP_OP » (DB99) qui contient la consigne pièces.
3. Ouvrez également la vue du Touchpanel «Convoyeur »
4. Fractionnez la fenêtre d'édition (voir figure).
5. Configurez le champ d'entrée pour la saisie de la consigne pièces en faisant glisser la variable « Consigne_pièces » du bloc de données vers la vue du Touchpanel.
6. Sélectionnez le champ d'entrée / de sortie nouvellement créé et dans la fenêtre Inspecteur de "Propriétés -> Général -> Format" définissez le "Modèle de format" (trois chiffres). En outre, vous pouvez ajuster comment tout à l'air dans les menus "Apparence" et "Format de texte".

12.7.5. Exercice 6: compiler et enregistrer le projet HMI



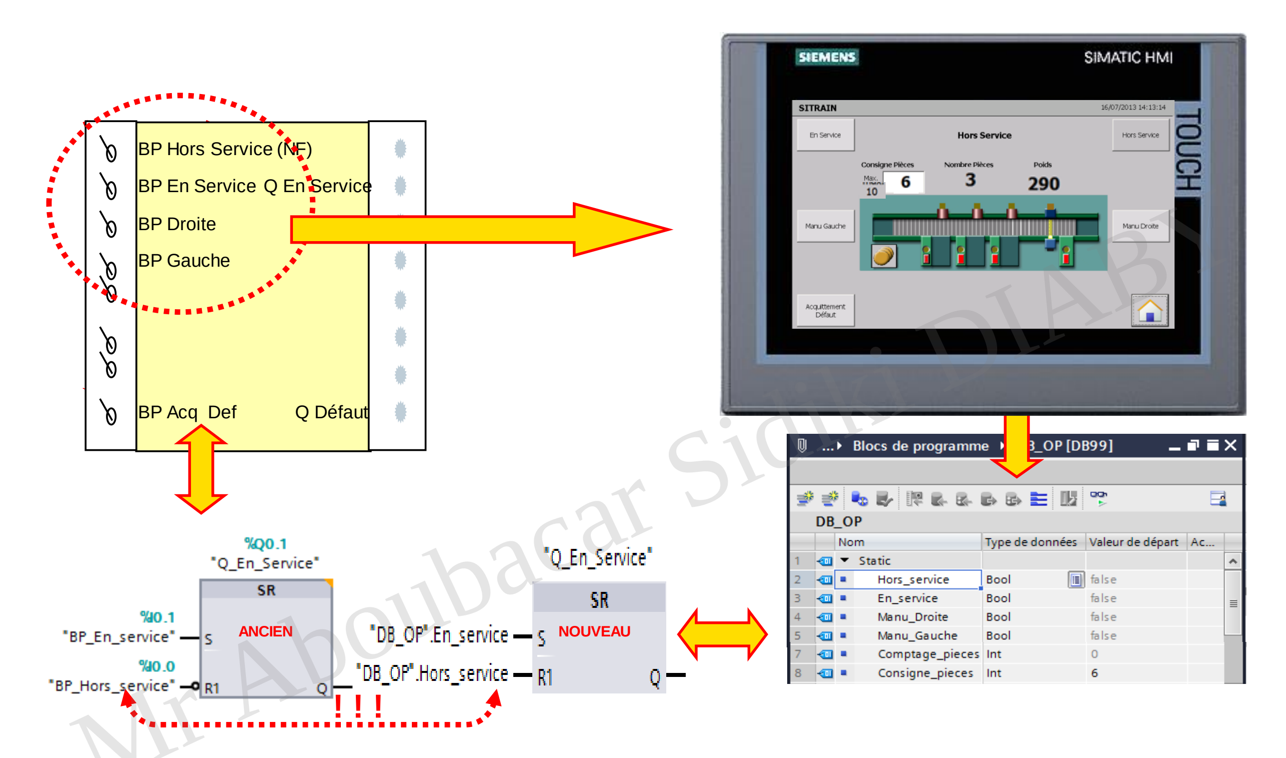
Enoncé

Vous devez à présent compiler et enregistrer le projet IHM terminé.

Marche à suivre

1. Compilez le projet IHM, en sélectionnant le Touchpanel dans le Navigateur du projet, puis en cliquant sur le bouton « Compiler » (voir figure).
2. Dans la fenêtre d'inspection sous « Info » visualisez les résultats de l'opération de compilation et éliminez les erreurs éventuellement apparues.
3. Enregistrez votre projet.

12.8. Exercice 7: Adapter le programme S7



Enoncé

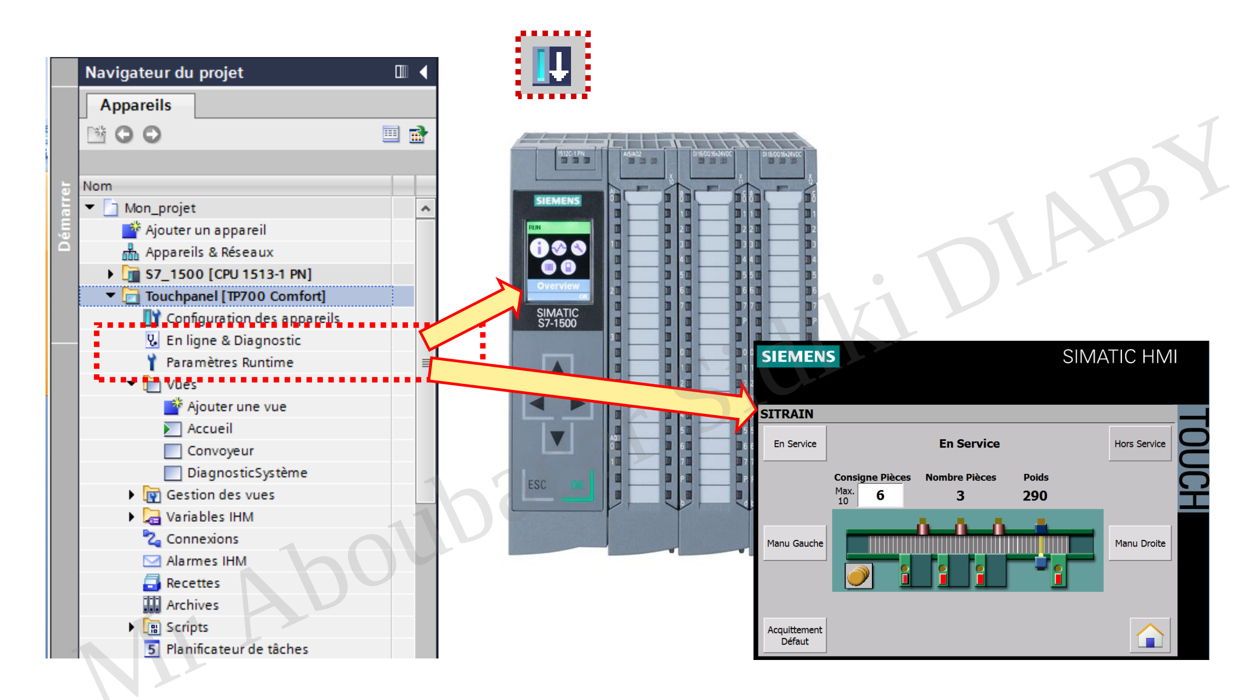
Mettez en service le projet du Touchpanel et adaptez le programme S7 de l'API de telle manière à ce que…

- Les fonctions En Service, Hors Service et Marche Manu_Droite / Gauche ne soient plus pilotées via les boutons poussoir du simulateur, mais via les touches correspondantes du Touchpanel.
- L'acquittement d'un défaut de fonctionnement de la bande soit possible non seulement comme jusqu'à présent, via le bouton poussoir du simulateur « BP_Acq_Def » (I0.7) mais encore via la touche correspondante du Touchpanel.
- La consigne pièces ne soit plus fixée à 3 mais puisse être saisie via un champ d'E/S du Touchpanel.

Marche à suivre

1. Remplacez dans le « FC_En_Service » (FC15) les variables « BP_En_Sercice » (I0.1) et « BP_Hors_Service » (I0.0) par les variables correspondantes du DB (voir figure). Tenez compte du fait que le bouton « BP_Hors _Service » soit câblé en tant que contact à ouverture
2. Complétez « OB_cyclique » (OB1) par l'appel du « FC_Défaut » (FC17) de telle sorte qu'un acquittement de défaut soit possible comme jusqu'à présent via « BP_Acq_Def » (I0.7) et de plus via le Touchpanel et la variable du DB « DB_OP.Acq_def_Tapis ».
3. Remplacez dans le « FC_Convoyeur » (FC16) les variables « BP_Droite »(I0.2) et « BP_Gauche »(I0.3) par les variables correspondantes du DB.
4. Vérifiez que la consigne des pièces puisse être saisie via le Touchpanel.
5. Vérifiez également toutes les autres fonctions et enregistrez votre projet.

12.8.1. Exercice 8: Chargement du projet IHM et CPU



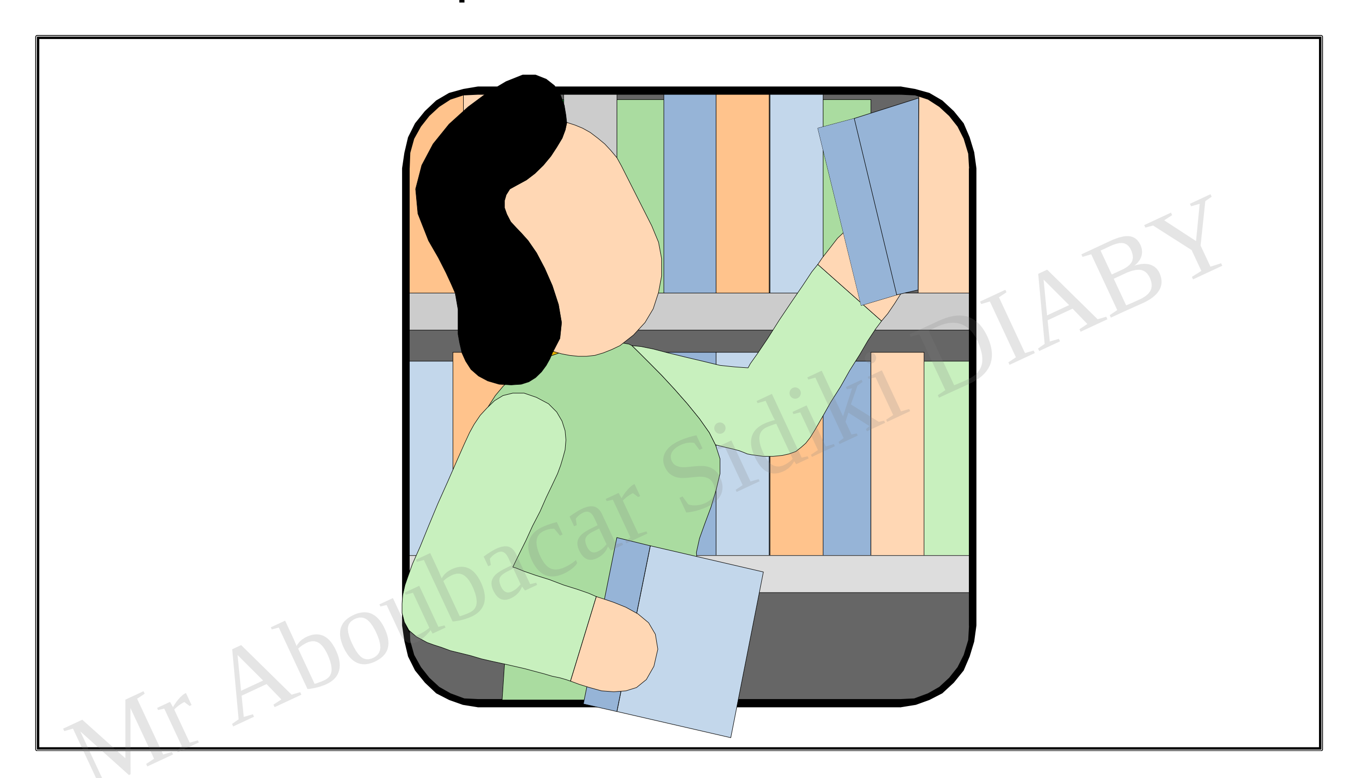
Enoncé

A partir du projet maintenant terminé, tous les blocs S7 doivent à nouveau être transférés dans la CPU et le projet entier du Panel doit être transféré dans l'écran tactile

Marche à suivre

1. Téléchargez tous les blocs S7 dans la CPU.
2. Téléchargez le projet Panel dans l'écran tactile.
3. Effectuer un test de fonctionnement par
   – Activation et désactivation de l'opération via l'écran tactile.
   – Lorsque l'opération est désactivée, déplacez le modèle de convoyeur vers la droite et vers la gauche à l'aide de l'écran tactile.
   – Lorsque l'opération est sur ON, préréglez une valeur de Consigne et commencez à transporter les pièces,
   – Surveillez le nb de pièces produites.
   – Cause un défaut de convoyeur, surveillez son affichage sur l'écran tactile et acquitter le défaut sur l'écran tactile
4. Corrigez - si nécessaire - votre programme et vérifiez son fonctionnement à nouveau.
5. Enregistrer le projet.

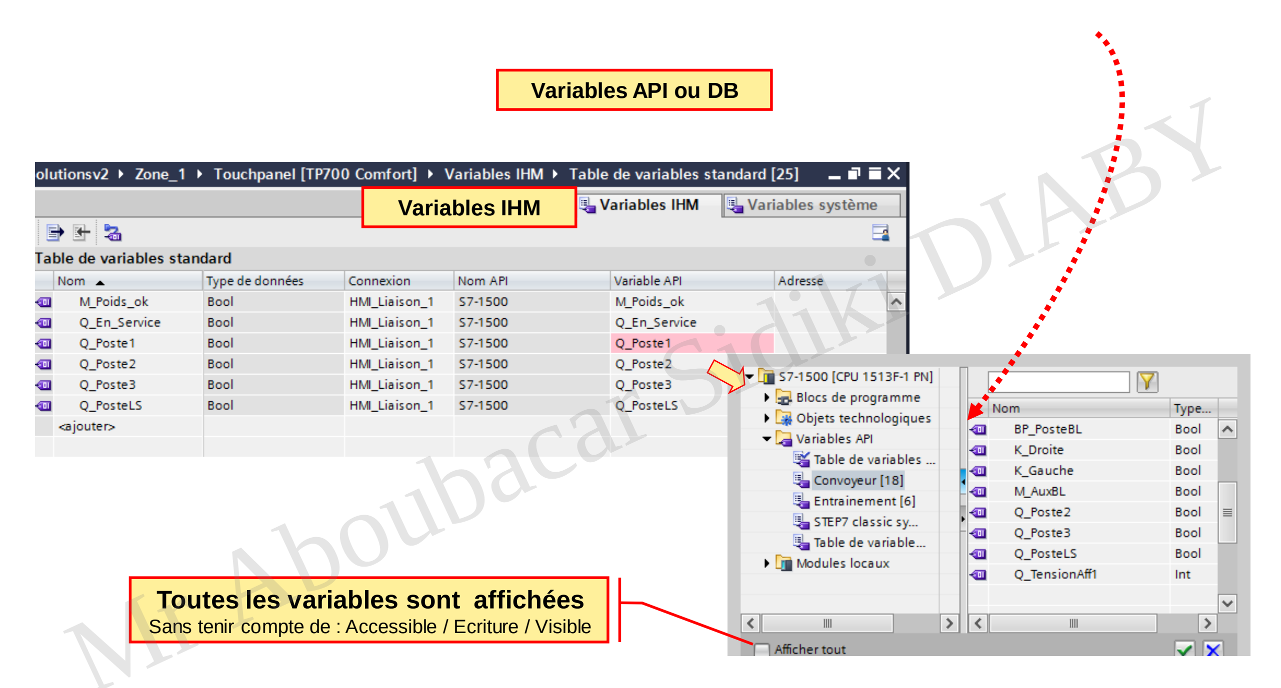
12.9. Information complémentaire



Commentaire

Les pages suivantes contiennent d'autres informations ou sont destinées à compléter un sujet.
Pour une étude plus approfondie, nous proposons des cours avancés et des supports d'auto-apprentissage.

12.9.1. Accès HMI / OPC UA aux variables API et aux variables de base de données



Accès au variable HMI

Dans les tableaux de variables API et les blocs de données, il est possible de déclarer les mécanismes de protection à l'aide desquels les accès non désirés aux variables API et aux variables DB des pupitres opérateur ou des UA-Clients OPC peuvent être évités:

• "Visible dans l'ingénierie IHM":
Pendant la configuration IHM, seules les variables / variables avec l'attribut "Visible dans HMI Engineering" peuvent être sélectionnées. Cette fonction de filtre peut cependant être désactivée dans la boîte de dialogue de sélection affichée en activant "Afficher tout".

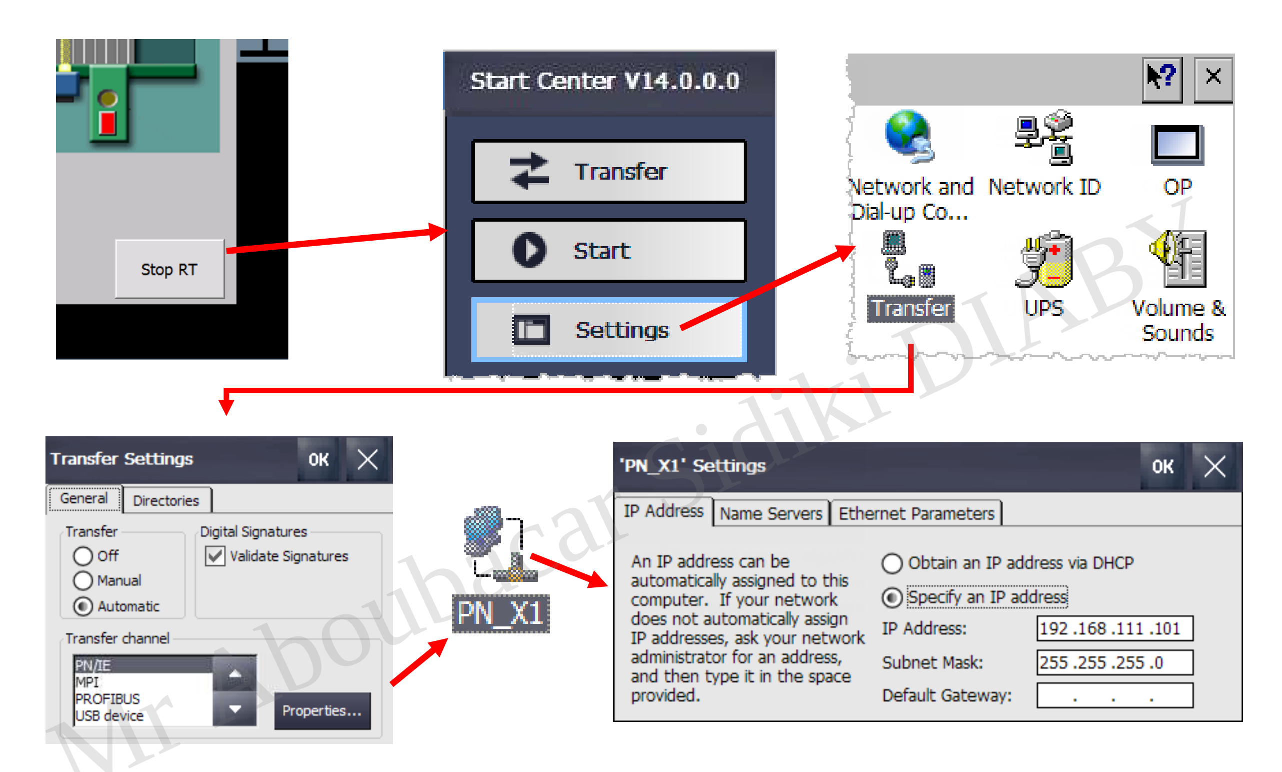
• "Lecture / écriture à partir de HMI / OPC UA" (uniquement S7-1200 et S7-1500):
Ceci indique si l'étiquette / la variable peut être écrite à partir de HMI / OPC UA à l'exécution. Cette fonction de protection intégrée au système d'exploitation S7-1200 / 1500 garantit que le pupitre opérateur ou le client UA OPC n'écrase pas certaines variables / variables.

• "Accessible depuis HMI / OPC UA" (uniquement S7-1200 et S7-1500):
Le pupitre opérateur ne peut accéder en ligne qu'aux variables / variables ayant l'attribut "Access from HMI / OPC UA". Cette fonction de protection intégrée dans le système d'exploitation S7-1200 / 1500 garantit que le pupitre opérateur ou le client OPC UA ne peuvent ni accéder en lecture ni accéder en écriture à certaines variables / variables. Les variables / variables qui ne sont pas "Accessibles à partir de HMI / OPC UA" ne sont donc pas "visibles dans l'ingénierie HMI".

Commentaire :

OPC UA est une norme qui permet l'échange de données indépendant du système d'exploitation.

12.9.2. Définition manuelle de l’adress IP sur le panneau



Définition manuelle de l'adresse IP sur le panneau

Vous pouvez définir manuellement l'adresse IP de l'interface du panneau via le Centre de démarrage> Paramètres.

Sommaire

13

13. Blocs d'organisation

A la fin du chapitre les participants doivent..



... être familier avec les différents type de blocs d'organisation

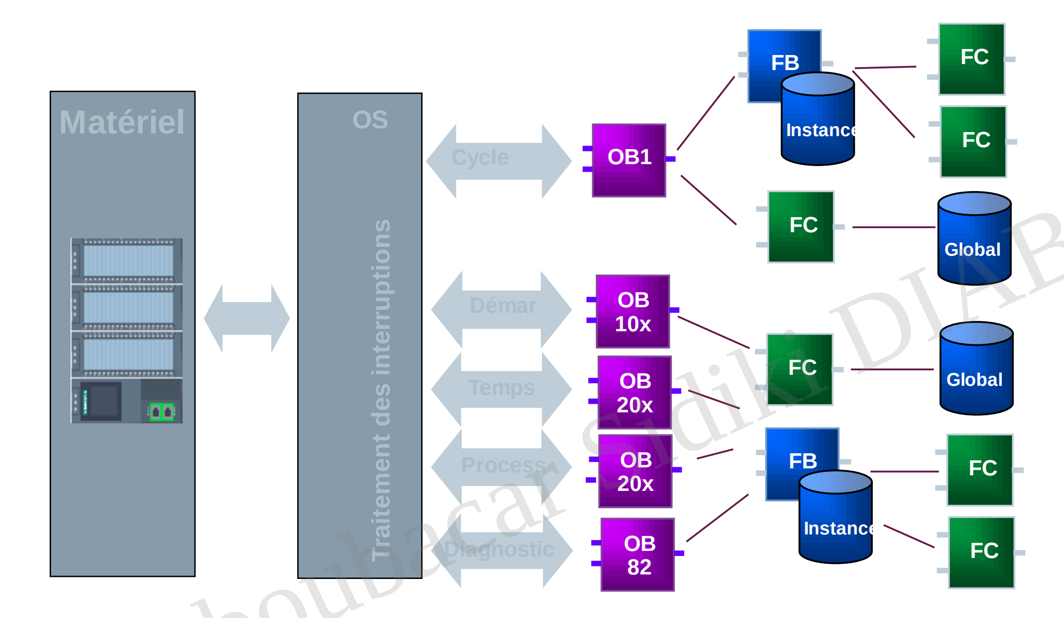
... comprendre les principaux processus d'interruption

... être familier avec la signification des mémoires images process

... être capable d'interprêté les informations de démarrage des OBs

… être capable d'utiliser les Obs de démarrage et d'interruption

13.1. Blocs d'organisation : Vue d'ensemble



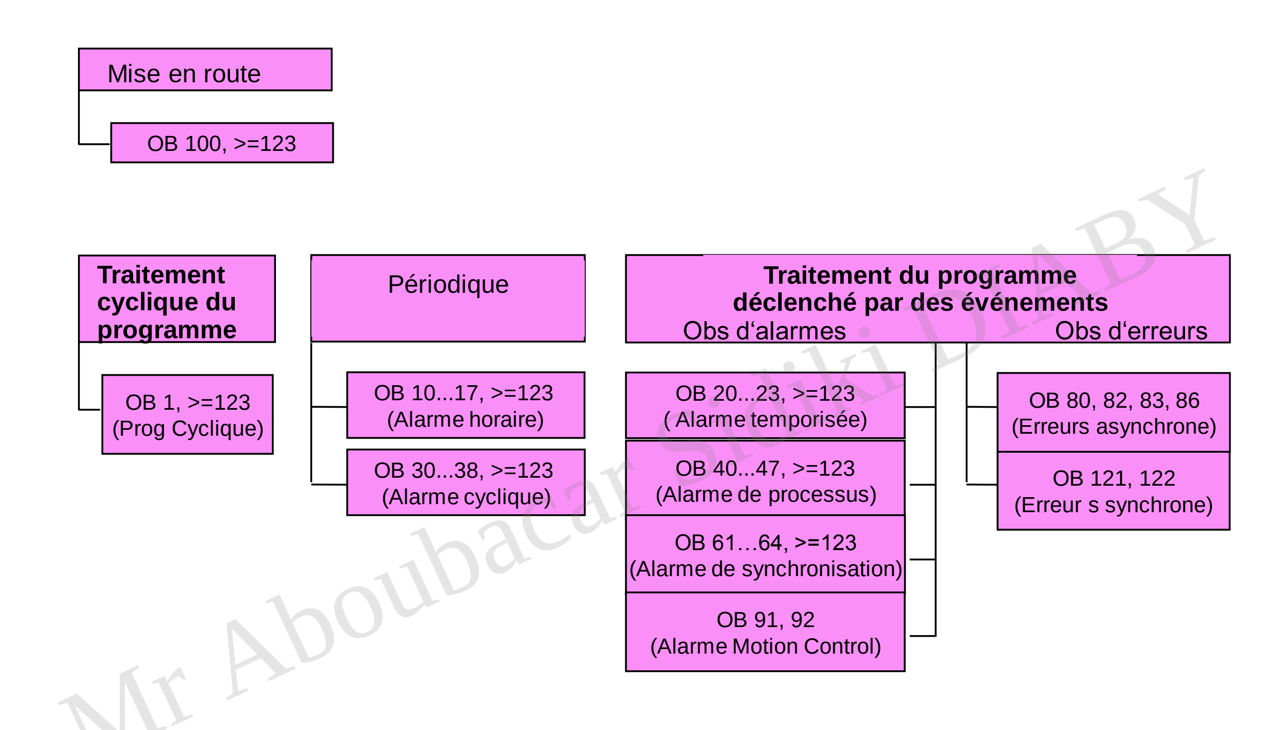
Blocs d’organisation (OBs)

Les blocs d'organisation (OB) forment l'interface entre le système d'exploitation et le programme utilisateur. Les blocs d'organisation sont appelés événements pilotés par le système d'exploitation.

Les événements peuvent être cycliques, la transition STOP-RUN, dépendant du temps, dépendante du matériel ou une erreur.

Si l'un des événements se produit, l'OB concerné est appelé s'il est chargé et si sa priorité est supérieure à celle de l'OB en cours de traitement.

13.1.1. Blocs d'organisation du S7-1500



Programme de mise en route

Après le rétablissement de la tension d'alimentation, un changement de mode de fonctionnement à l'aide du sélecteur de mode situé sur la CPU ou encore une commande de démarrage lancée sur la console de programmation, le système exécute un programme de mise en route avant d'entamer le traitement cyclique du programme. Les liaisons de communication peuvent être établies avec un paramétrage par défaut dans les blocs de démarrage.

Traitement cyclique du programme

Le programme mémorisé dans le bloc d'organisation OB 1 est traité en continu de manière cyclique. Pour le traitement cyclique du programme, le temps de cycle est constitué du temps de traitement pour le système d'exploitation de la CPU et de la somme des temps d'exécution de toutes les instructions du programme. Le temps de réaction, à savoir la vitesse avec laquelle une sortie peut être commutée après l'émission d'un signal d'entrée, est égal au minium à 1 fois le temps de cycle, et au maximum à deux fois le temps de cycle.

Traitement périodique du programme

Le traitement cyclique du programme peut être interrompu à intervalles de temps réguliers. Dans le cas des alarmes cycliques, un bloc d'organisation (par ex. l'OB 35) est traité après écoulement d'une base de temps paramétrable (par exemple toutes les 100 ms). Ces blocs d'organisation peuvent appeler, par exemple, des blocs de régulation avec leur temps de scrutation. Dans le cas des alarmes horaires, un OB est traité à une heure déterminée, par exemple chaque jour à 17h00 pour lancer une sauvegarde des données

Traitement du programme à déclenchement événementiel

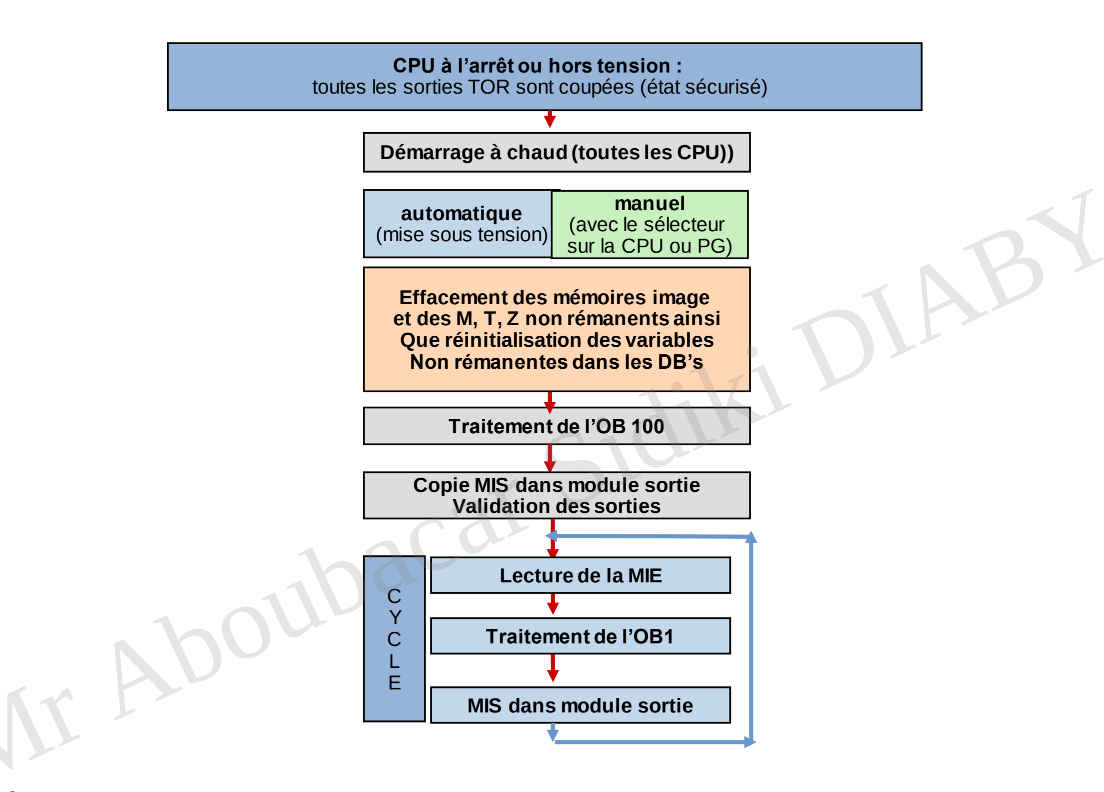
Les alarmes de processus permettent de réagir rapidement aux événements survenant dans le processus. Dès que l'événement apparaît, le cycle est interrompu et un programme d'alarme est traité. L'alarme temporisée réagit à un événement dans le processus après un certain délai. Les OB d'erreur déterminent le comportement de l'automate en cas d'erreur logicielle ou de défaut matériel.

Traitement du programme en tâche de fond

Certaines CPU S7 permettent de définir un temps de cycle minimum. Si le temps de traitement pour les opérations système et les programmes de l'OB1 et des alarmes est au total inférieur au

temps de cycle minium convenu, le programme contenu dans l'OB de traitement en arrière-plan est traité pendant le temps restant.

13.2. S7-1500 Sequence de démarrage et cyclique



General

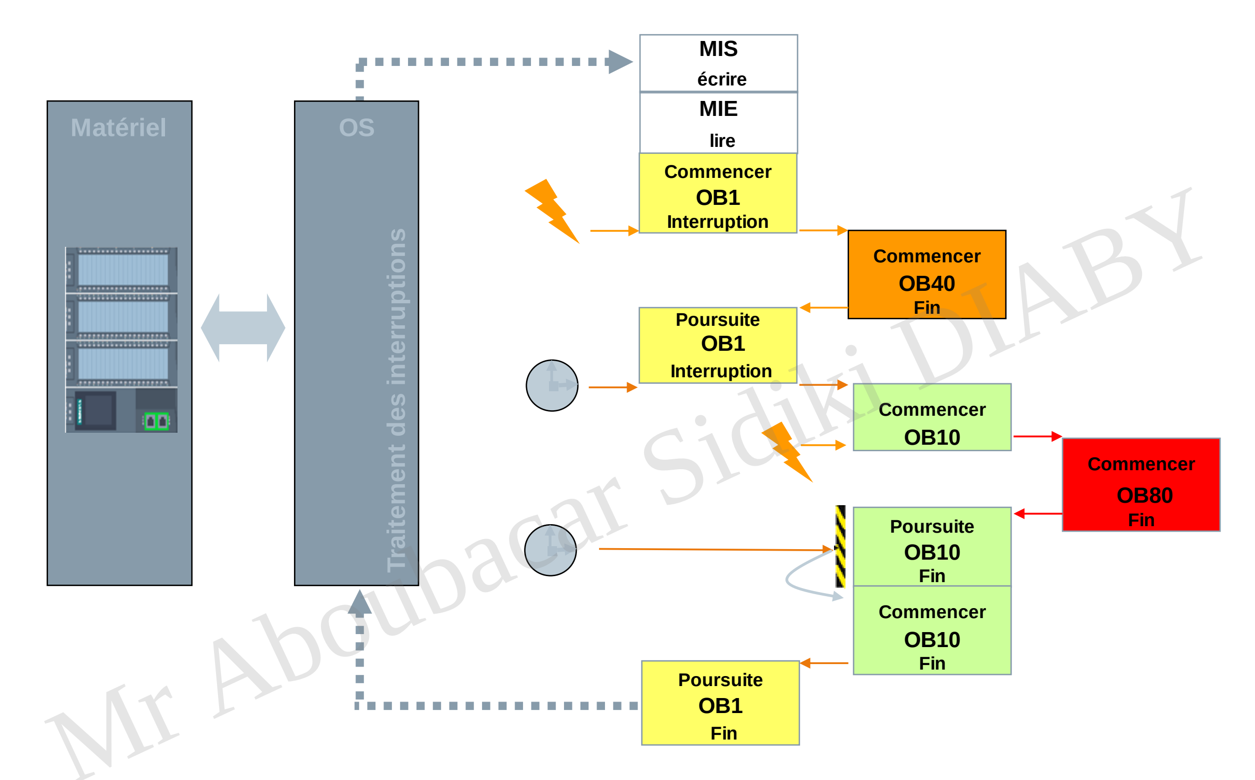
Avant que la CPU ne commence à exécuter le programme utilisateur cyclique, un programme de démarrage est exécuté.

Dans le programme de démarrage, les variables d'initialisation peuvent être définies en programmant les OB de démarrage en conséquence

Redémarrage à chaud

Le S7-1500 effectue un redémarrage à chaud dans lequel les images de process (PII, PIQ) et tous les bits de mémoire, timers et compteurs non rémanents sont supprimés. Les DB non rémanents sont réinitialisés aux valeurs initiales de la mémoire de chargement et les bits de mémoire rémanents, les temporisations et les compteurs ainsi que les contenus DB rémanents sont conservés.

13.2.1. Interruption du programme cyclique



Appel d'OB

Les blocs d'organisation (OB) forment l'interface entre le système d'exploitation de la CPU et le programme utilisateur.

Les blocs d'organisation sont appelés exclusivement par le système d'exploitation. Il y a différents événements de démarrage (interruptions de l'heure du jour, interruptions matérielles - voir l'image) qui conduisent chacun au début de leur bloc d'organisation associé.

Interruption du programme cycliqueInterrupting the Cyclic Program

Lorsque le système d'exploitation appelle un autre OB, il interrompt l'exécution du programme cyclique car un "cycle de programme" -OB a la priorité la plus basse. Tout autre OB peut donc interrompre le programme principal et exécuter son propre programme. Ensuite, le "Cycle du programme" -OB reprend l'exécution au point d'interruption.

Priorités

Les CPU S7-1500 prennent en charge les priorités 1 (priorité la plus basse) à 26 (priorité la plus élevée).

Les OB sont exécutés sur une base purement prioritaire. Cela signifie que lorsque plusieurs demandes d'OB existent en même temps, l'OB ayant la plus haute priorité est exécuté en premier. Lorsqu'un événement ayant une priorité plus élevée que l'OB actif est présent, cet OB est interrompu. Les événements de même priorité sont exécutés dans l'ordre où ils se produisent.

Si c'est également le cas, par exemple pour les OB de démarrage, les OB sont exécutés en fonction de leur numéro.

13.2.2. Mémoire image processus – complète et partiel

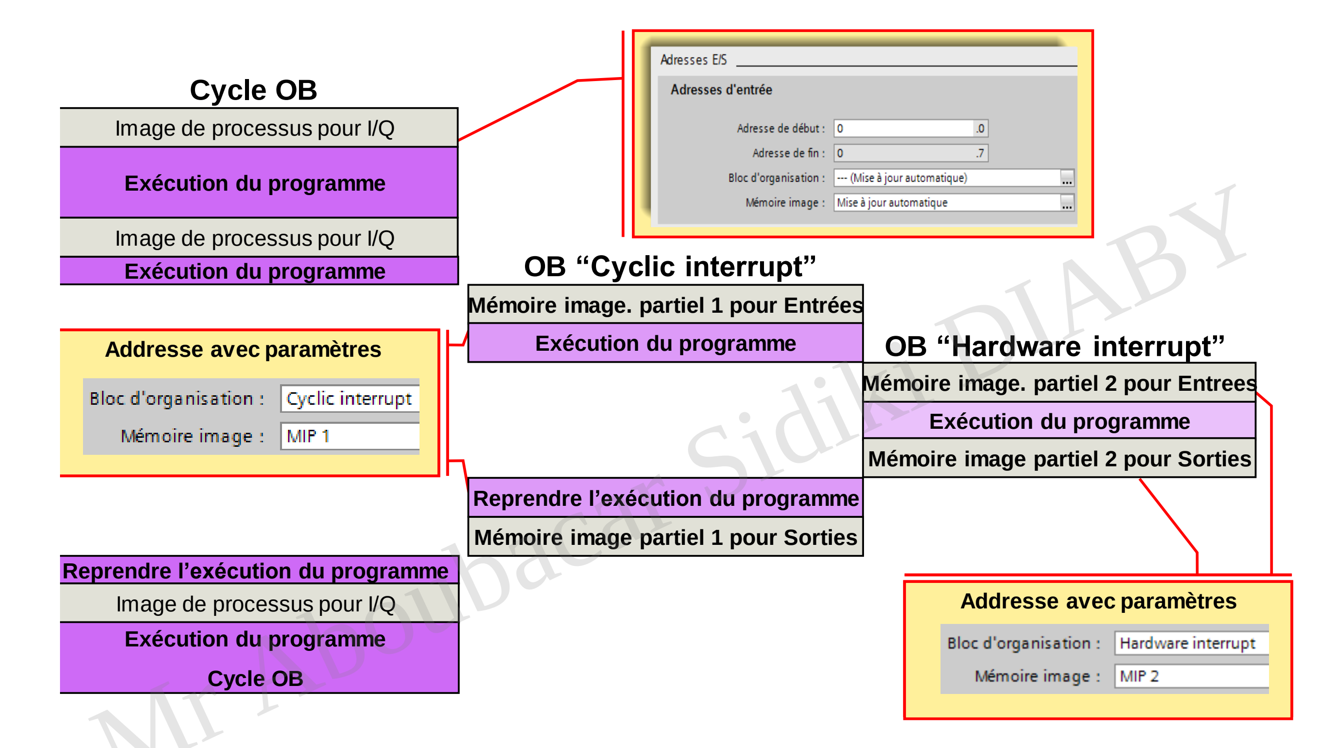


Image de processus pour les entrées et les sorties

Pour des raisons de vitesse d'accès et de cohérence de l'état des entrées individuelles tout au long du cycle, une copie est stockée dans l'image de process pour les entrées (MIE) auxquelles on accède pendant l'exécution du programme. L'écriture des sorties se produit dans l'image de process pour les sorties (MIS). Les états des sorties stockées ici sont écrits dans les sorties réelles au début de chaque cycle.

Les entrées et les sorties d'un S7-1500 sont automatiquement mises à disposition dans l'image de process pour les entrées et les sorties tant que cela n'est pas modifié dans les réglages des blocs individuels.

Sous les propriétés du module> Général> Entrée ... / Sortie ...> Adresses d'E / S, "Mise à jour automatique est défini dans le bloc Organisation et l'image de processus.

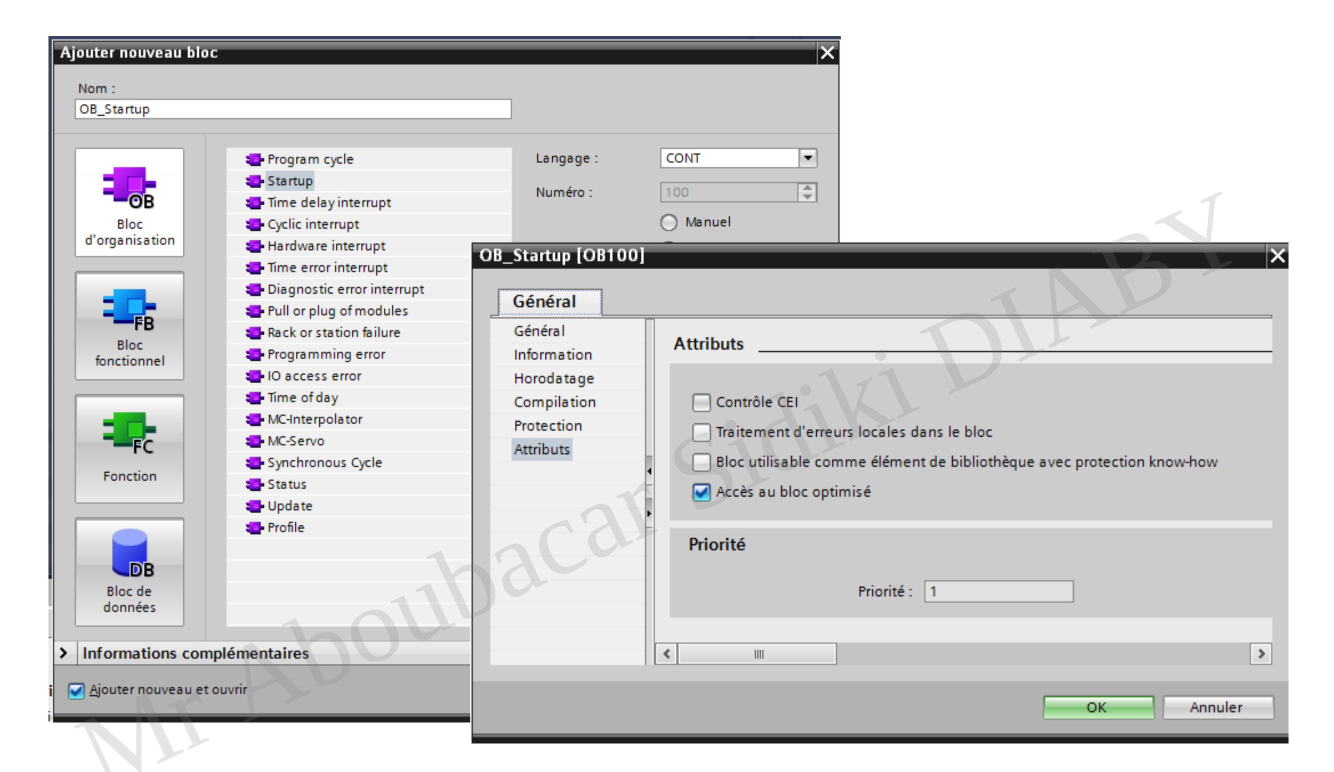
Image du Processus partielle (MIP)

Les entrées et les sorties qui ne sont pas utilisées dans le programme cyclique, c'est-à-dire pas dans chaque cycle, n'ont pas besoin d'être mises à jour avec l'image de process au début de chaque cycle. Pour cette raison, il existe des partitions d'image de processus qui, à leur tour, peuvent être affectées à des blocs d'organisation individuels. Si un OB auquel est attribué un MIP est exécuté, les entrées du MIP associé sont lues au début et à la fin, les états des sorties du MIP sont écrits dans les périphériques correspondants. Si aucune partition d'image de processus n'est affectée à un bloc d'organisation, la mise à jour de ce MIP doit être effectuée à l'aide d'instructions.

Mise à jour d'une partition d'image de processus dans le programme utilisateur

Chaque MIP peut être mis à jour dans le programme utilisateur avec des instructions spéciales. Pour cela, les instructions "UPDAT_PI" pour la mise à jour des entrées et "UPDAT_PO" pour la mise à jour des sorties sont utilisées.

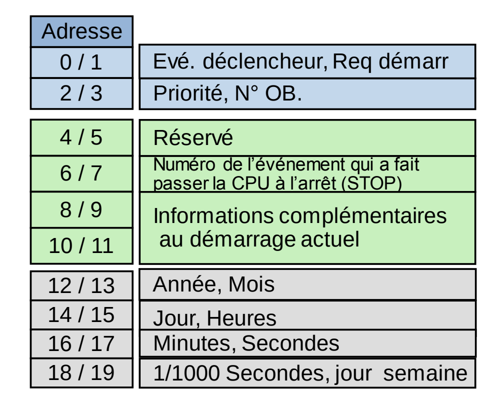
13.3. Créer un nouvelle OB



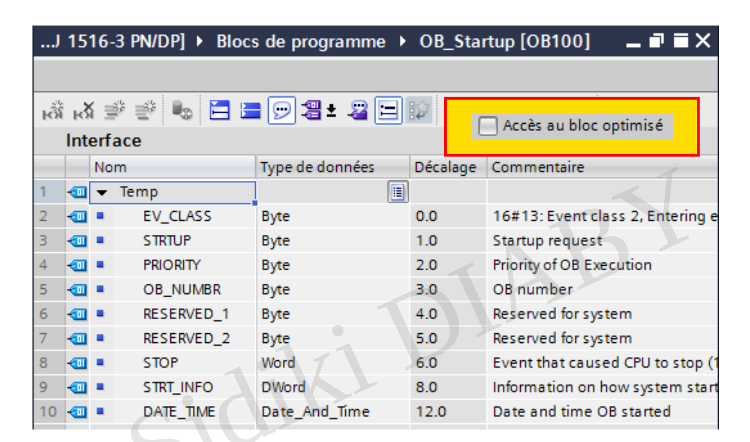
Créer un nouvelle OB

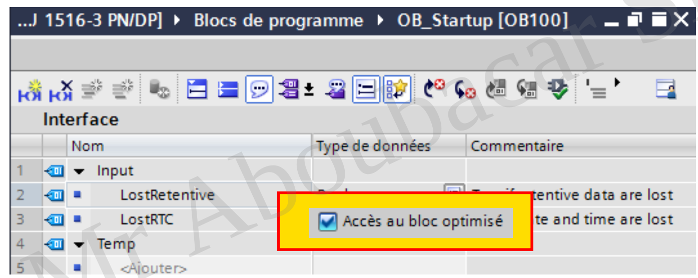
Lors de la création d'un bloc d'organisation, le type d'événement est tout d'abord sélectionné. En outre, le numéro et le nom peuvent être modifiés. Par défaut, les nouveaux OB sont créés avec l'attribut "Accès optimisé au bloc", de sorte que seules les informations de démarrage réduites sont disponibles dans les OB (voir page suivante). Dans les propriétés de l'OB, le comportement peut être défini individuellement en fonction du type.

13.3.1. OB Information de départ en utilisant "OB_Startup" comme exemple



Adresse	
0 / 1	Evé. déclencheur, Req démarr
2 / 3	Priorité, N° OB.
4 / 5	Réservé
6 / 7	Numéro de l'événement qui a fait passer la CPU à l'arrêt (STOP)
8 / 9	Informations complémentaires au démarrage actuel
10 / 11	
12 / 13	Année, Mois
14 / 15	Jour, Heures
16 / 17	Minutes, Secondes
18 / 19	1/1000 Secondes, jour semaine





Informations de départ pour l'accès au bloc non optimisé

Lorsque le système d'exploitation appelle des blocs d'organisation, l'utilisateur reçoit des informations de départ spécifiques à l'OB sur la pile de données locale.

Ces informations de départ ont une longueur de 20 octets et sont disponibles après le début de l'exécution de l'OB.

Les informations de départ ainsi que leurs adresses Pile-L absolues ne sont complètement disponibles que pour les OB où l'attribut de bloc "Accès optimisé au bloc" **n'est pas activé** (comme indiqué sur l'image).

Afin d'éviter les erreurs, la structure de la section de déclaration standard ne doit pas être modifiée par l'utilisateur. Après la section de déclaration standard, l'utilisateur peut déclarer d'autres variables temporaires supplémentaires.

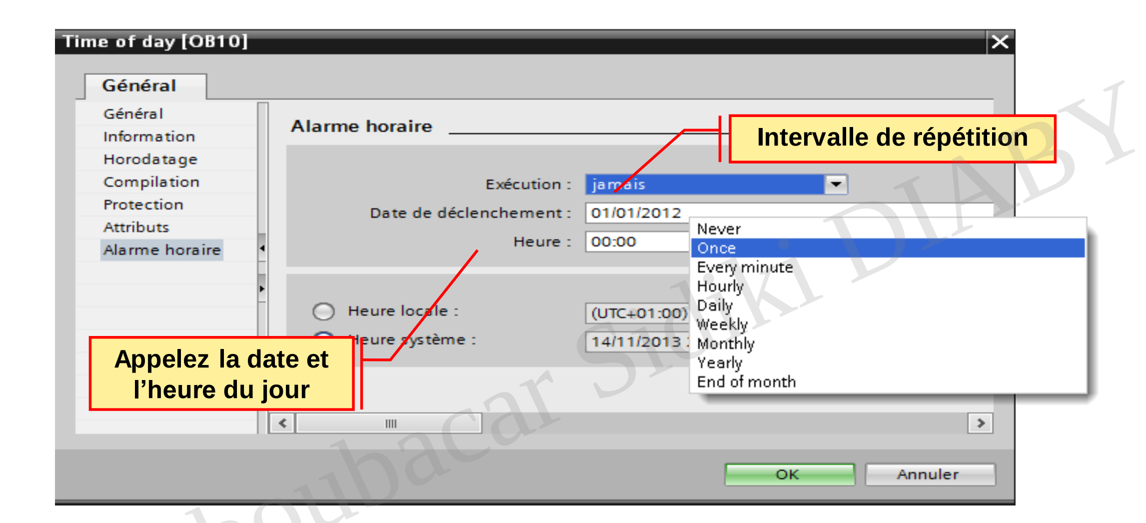
Informations de départ pour l'accès au bloc optimisé

Les informations de départ d'un bloc d'organisation avec un accès au bloc optimisé sont limitées à l'essentiel et sont transmises au moyen de paramètres d'entrée. Si nécessaire, les informations de départ, à l'exception de la date et de l'heure, peuvent être lues à l'aide de la fonction "RD-SINFO".

Variables

Une explication de la signification des variables peut être trouvée dans l'aide en ligne.

13.4. Alarme horaire



Alarmes horaires

Les alarmes horaires sont utilisées pour exécuter un programme donné, appelé dans l'OB 10, une seule fois à un moment précis ou périodiquement à partir de ce moment (toutes les minutes, toutes les heures, tous les jours, toutes les semaines, tous les mois, déclenchement annuel).

En fonction de la CPU, l'utilisateur dispose au maximum de huit alarmes horaires différentes.

Activé

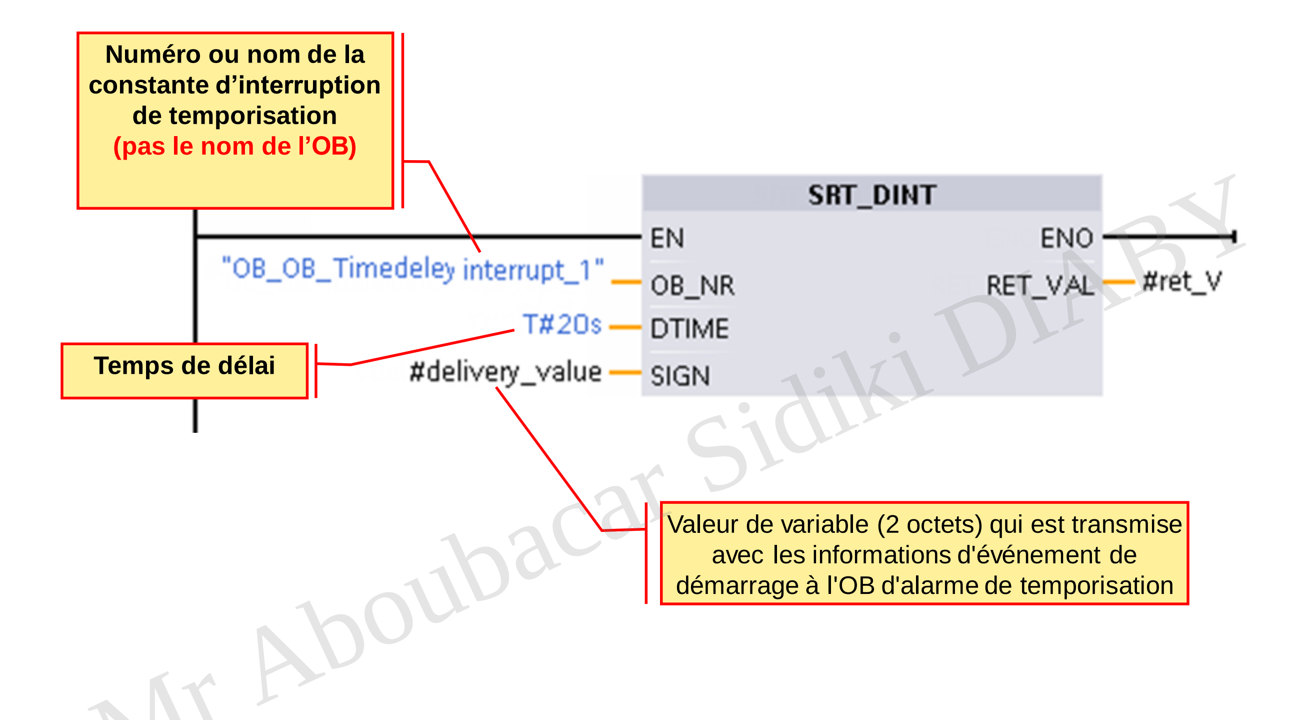
Lorsque l'option « Activé » est cochée, l'OB d'alarme horaire est automatiquement lancé à chaque démarrage de la CPU.

Comentaire

Les alarmes horaires peuvent également être commandées en cours d'exécution avec les « Instructions étendues -> Alarmes -> Interruption de l'heure » ci-dessous :

- "SET_TINT" Réglage de la date de déclenchement, de l'heure et de la période
- "SET_TINTL" Réglage de la date de déclenchement, de l'heure et de la période
- "CAN_TINT" Suppression de l'alarme horaire
- "ACT_TINT" Activation de l'alarme horaire
- "QRY_TINT" Interrogation de l'alarme horaire

13.4.1. Démarrer l'OB d'interruption de temporisation



Interruption de temporisation

Les OB d'alarme de temporisation sont utilisés pour pouvoir réagir à des événements librement définissables après une temporisation. Avec la fonction "SRT_DINT", vous définissez quelle interruption temporisée après expiration de l'heure doit être appelée par le système d'exploitation. A l'aide du paramètre d'entrée "SIGN", une valeur de la taille d'un mot peut être transmise à l'OB d'alarme de temporisation.

Attention :

En tant que paramètre réel du paramètre formel "OB_NR", le numéro d'OB ou le nom de constante de l'OB doit être spécifié. Le nom du symbole de l'OB à appeler n'est pas acceptable.

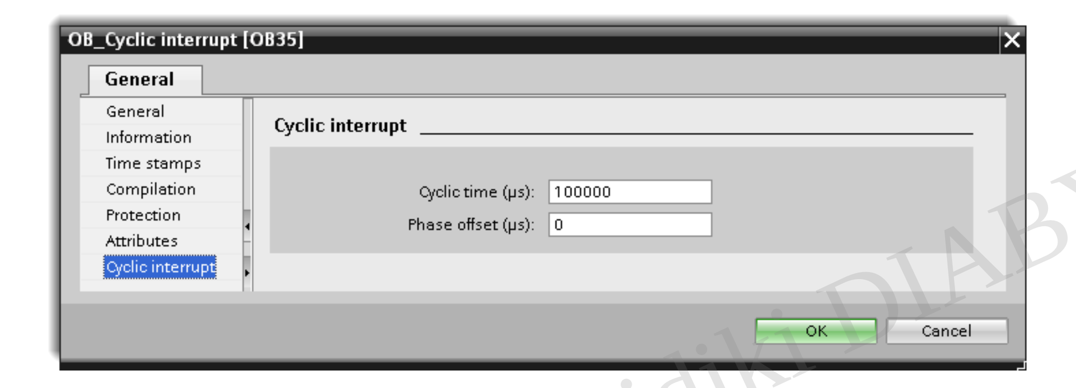
Vous allez remonter le nom de la constante dans les propriétés de l'OB sous "Général> Nom de la constante".

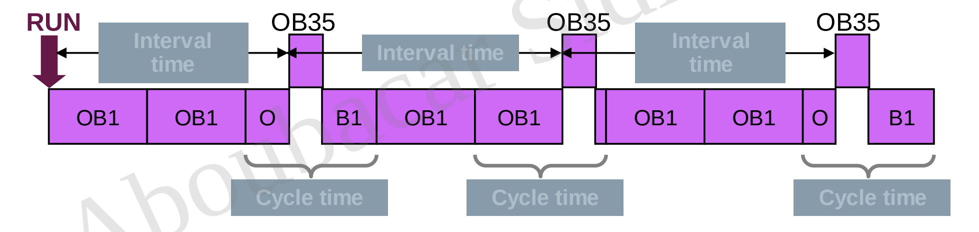
Remarque

En plus de l'instruction "RSD_DINT", il y a également d'autres instructions dans les instructions de la carte de tâche sous "Instructions avancées -> Interruptions> Instruction de temporisation":

- "CAN_DINT" Annuler l'interruption de temporisation
- "QRY_DINT" Interrogation de l'état de l'interruption de temporisation

13.4.2. Alarme cyclique





Alarme cyclique

Une alarme cyclique permet de lancer le traitement d’un bloc à intervalles réguliers. Le S7-1500 dispose de plusieurs OB d’alarme cyclique. Par défaut, l’intervalle de temps pour l’appel de l’OB est de 100 ms, la plage de réglage allant de 1 µs à 1 min.

Intervalle de temps

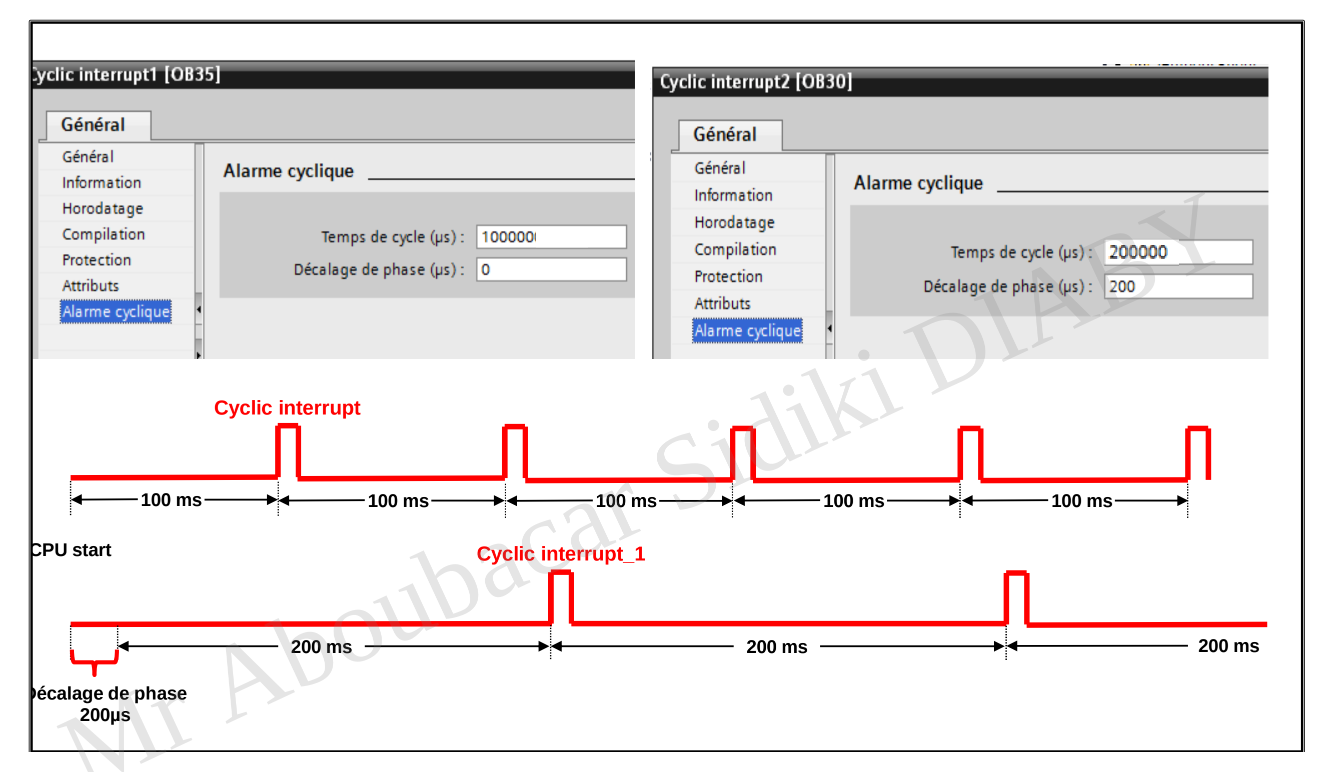
Il faut veiller à ce que l’intervalle défini soit supérieur au temps nécessaire à l’exécution. Le système d’exploitation appelle l’OB cyclique au moment indiqué ; si l’OB cyclique est encore actif à cet instant, le système d’exploitation appelle l'OB80 (erreur d’alarme cyclique).

Commentaire

Les interruptions cycliques peuvent également être contrôlées et interrogées au moment de l'exécution avec "Instructions étendues" (Instructions de la carte de tâche "Instructions étendues -> Interruptions> Interruption cyclique"):

- "SET_CINT" Définir les paramètres d'interruption cyclique
- "QRY_CINT" Interroger les paramètres d'interruption cyclique

13.4.2.1. Décalage de phase pour alarmes cycliques



Décalage de phase

Avec les OB d'alarme cyclique, vous pouvez démarrer des programmes à intervalles de temps réguliers. Pour ce faire, vous devez indiquer pour chaque
OB d'alarme utilisé un temps de cycle et un décalage de phase

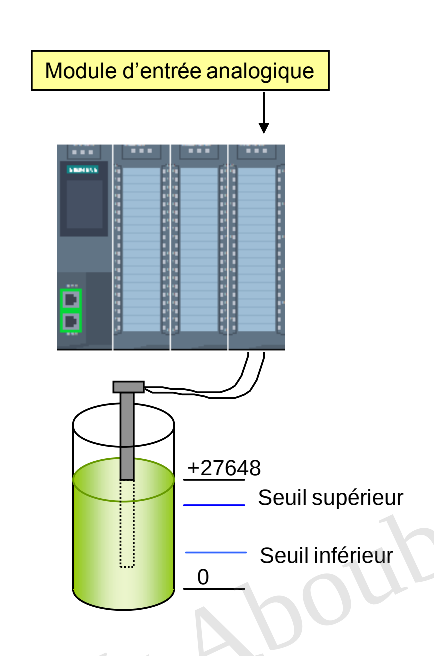
.

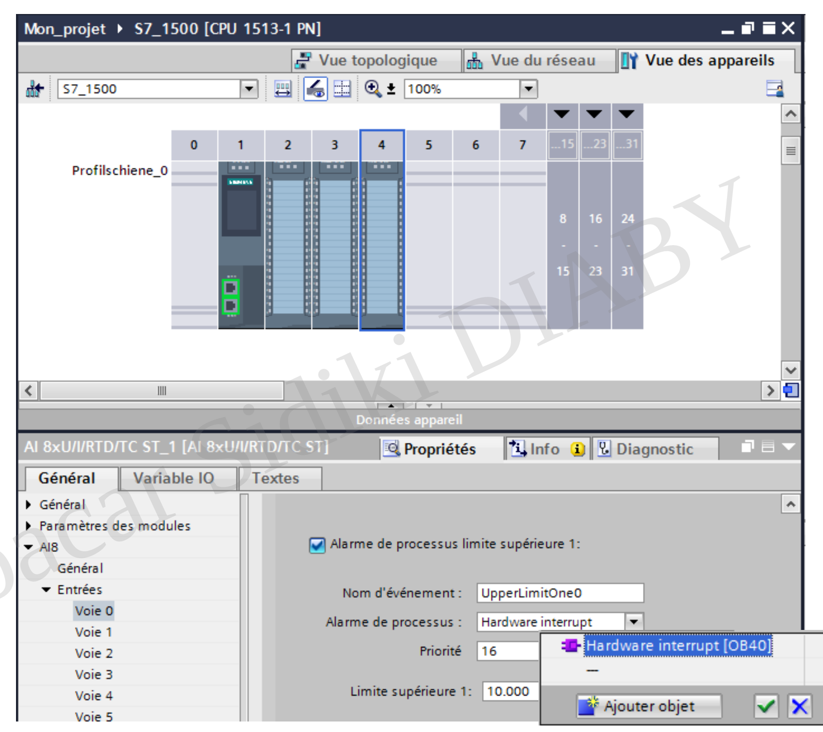
Nota Lorsque vous paramétrez plusieurs OB d'alarme cyclique, vous devez donner à chacun d'eux un autre temps de cycle ou un autre décalage de phase, afin d'éviter une exécution simultanée, ou une file d'attente. Lors de la création d'un OB d'alarme cyclique, les valeurs initiales sont un temps de cycle de 100 et un décalage de phase de 0.

Marche à suivre Pour saisir la période et le décalage de phase d'un OB d'alarme cyclique, procédez de la manière suivante :

- Dans le navigateur du projet, ouvrez le dossier "Blocs de programme".
- Cliquez avec le bouton droit de la souris sur OB D'alarme cyclique.
- Sélectionnez dans le menu contextuel la commande "Propriétés".
- La boîte de dialogue "<Nom de l'OB d'alarme cyclique>" s'affiche.
- Sélectionnez le groupe "Alarme cyclique" dans le navigateur.
- Les champs de saisie du temps de cycle et du décalage de phase s'affichent.
- Entrez le temps de cycle et le décalage de phase.
- Validez vos entrées avec "OK".

13.4.3. Alarme de processus





Alarme de processus

Le traitement du programme d'un OB d'alarme de processus est lancé dès qu'un événement déterminé survient dans le processus.

Les alarmes de processus peuvent être déclenchées par différents signaux provenant des modules :

- Sur les modules de signaux paramétrables (DI, DO, AI, AO), le signal qui doit déclencher l'alarme de processus est défini avec l'outil de configuration matérielle.
- Sur les modules CP (processeur de communication) et FM (module de fonction), le comportement de l'alarme est défini à l'aide du logiciel de configuration du module correspondant.

Exemple

Lors de la configuration d'un module d'entrée analogique, des valeurs limites appropriées ont été spécifiées dans l'exemple ci-dessus. Si la valeur mesurée dépasse alors cette limite, une interruption est déclenchée sur la CPU qui provoque l'interruption du programme et l'exécution de l'OB "Interruption du matériel".

13.5. Enoncé



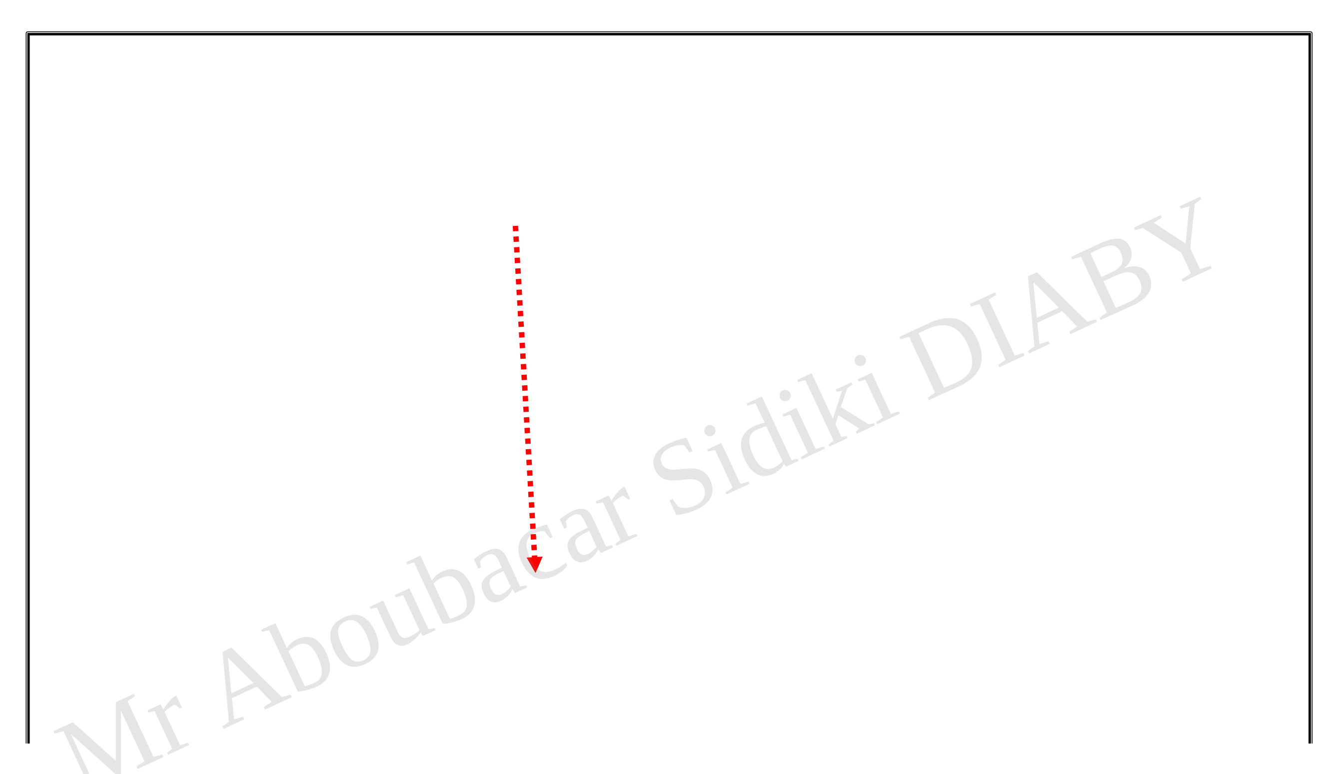


Enoncé

Afin d'éviter que des pièces ne se trouvent sur le convoyeur de transport après l'allumage ou le redémarrage à chaud, le convoyeur doit se déplacer vers la droite pendant 5 secondes lorsqu'il y a une transition STOP-RUN.

Pour cela, le convoyeur est démarré à l'aide d'un OB Startup pendant le démarrage et est de nouveau arrêté à l'aide d'un OB d'alarme de temporisation qui est appelé après 5 secondes.

13.5.1. Exercice 1: Préparation de l'initialisation au démarrage



Enoncé

Pour que le convoyeur se déplace vers la droite après le démarrage, il faut une variable qui a le statut Vrai pendant le temps d'initialisation.

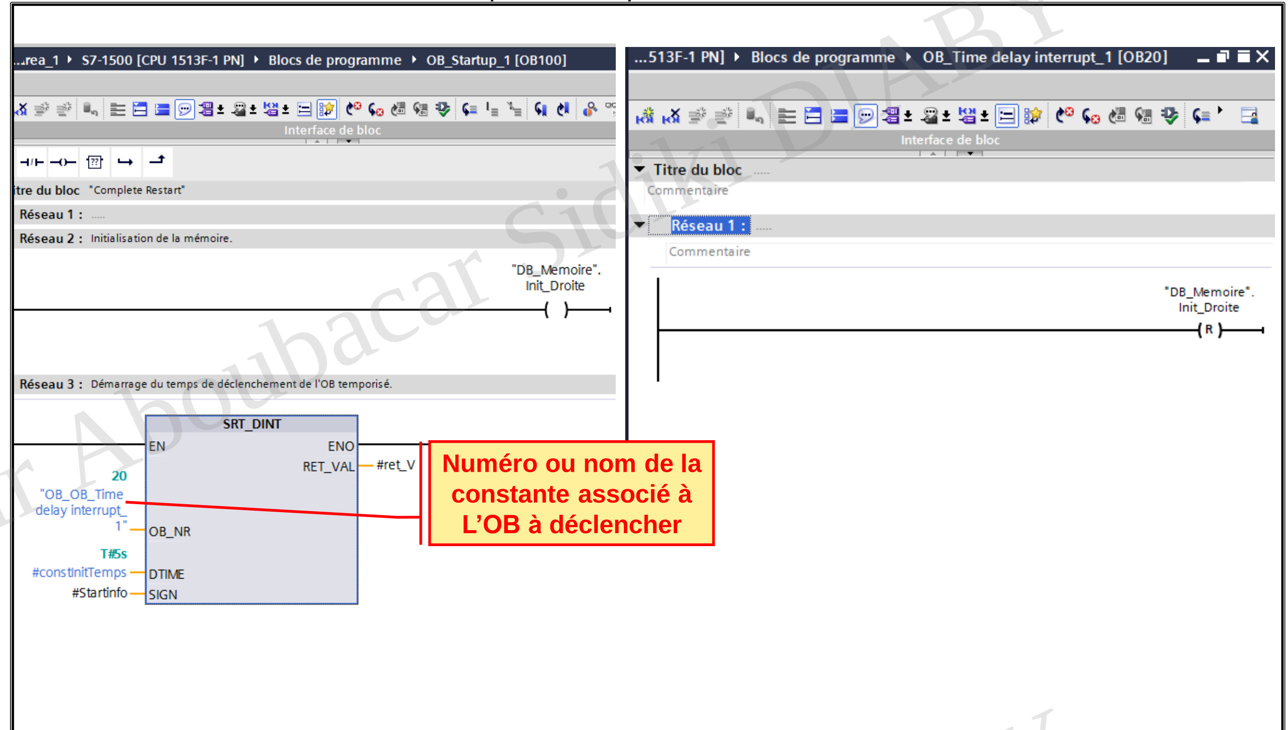
Marche à suivre

1. Insérez le nouveau bloc de données "DB_Mémoire" et déclarez la variable "Init_Droite".
2. Liez la variable "DB_Mémoire". Init_Droite dans "FC_Convoyeur" en tant que condition OU supplémentaire pour le mouvement du convoyeur de transport vers la droite.
3. Sauvegardez votre projet.

13.5.2. Exercice 2: Initialisation du transport en utilisant le démarrage et la programmation de l'OB d'alarme de temporisation

Enoncé

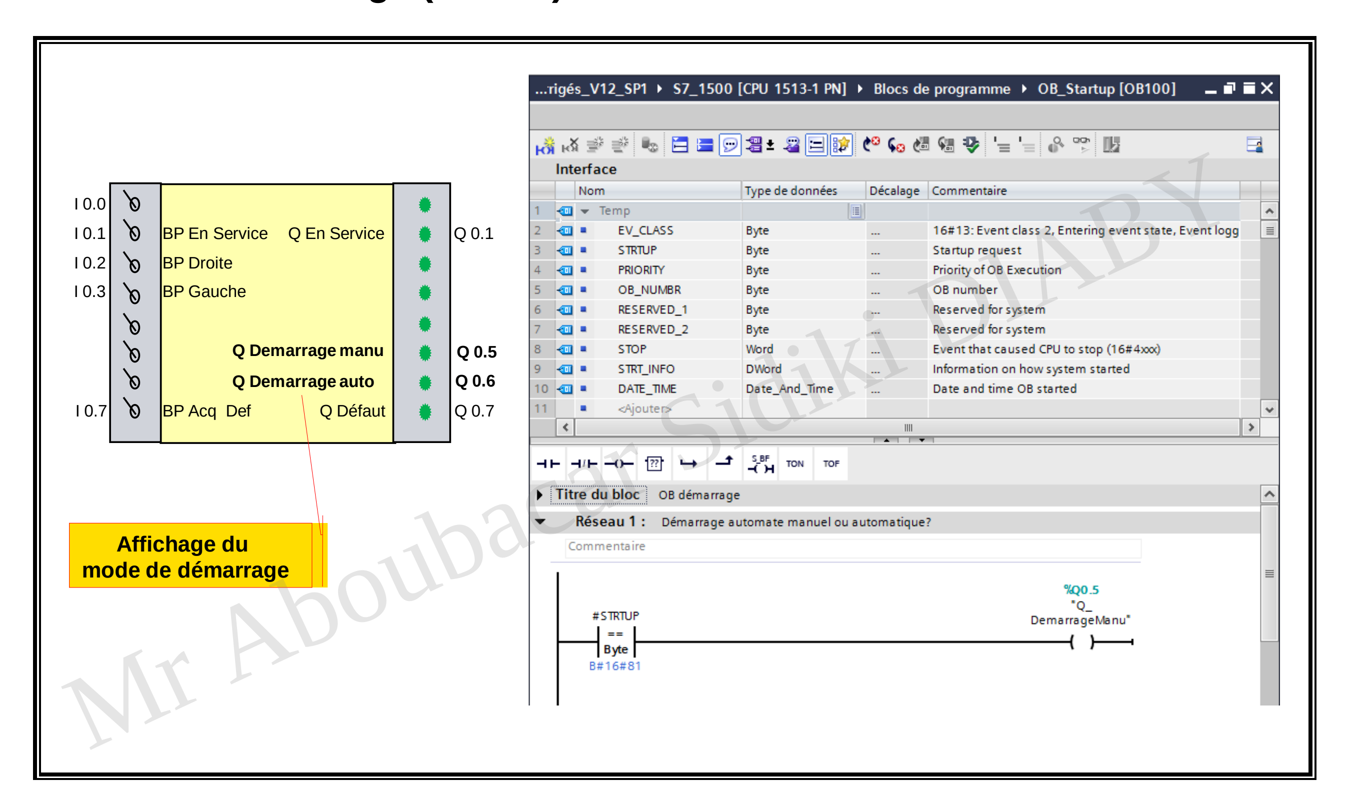
La nouvelle variable "DB_Memoire" . Init_Droite doit être définie dans un OB de démarrage et réinitialisée dans un OB d'alarme de temporisation après 5 secondes.



Marche à suivre

1. Insérez les nouveaux blocs d'organisation "OB_Startup_1" de type Startup et "OB_Time delay interrupt_1" du type Interruption temporisée.
2. Dans l'OB "OB_Startup_1", attribuez une affectation à la variable " DB_Memoire" . Init_Droite pour que la valeur TRUE lui soit affectée.
3. Dans un autre réseau, appelez la fonction "SRT_DINT" et déclarez-le de manière à ce que l'OB "Interruption de temporisation_1" soit démarré après 5 secondes.
   Attention: Le paramètre OB_NR n'accepte que le numéro de l'OB ou le nom de la constante de l'OB et non le nom du symbole.
4. Pour pouvoir fournir les paramètres SIGN et RET_VAL avec les paramètres réels, déclarez la variable temporaire "#ret_V" (INT).
5. Dans "OB_Time delay interrupt_1", réinitialisez la variable DB_Memoire" . Init_Droite.
6. Enregistrez, compilez et charger votre programme.
7. Testez la nouvelle fonction.

13.6. Exercice additionnel 2 : Visualisation et acquittement du mode de démarrage (OB100)



Enoncé

Il s'agit de signaler un **démarrage manuel** sur la LED du simulateur « *Demarrage_manuel* » (Q 0.5) et un **démarrage automatique** sur la LED du simulateur « *Demarrage_auto* » (Q 0.6).

Les deux LED doivent être désactivées dès lors que « Q_En_Service » (Q 0.1) est activé.

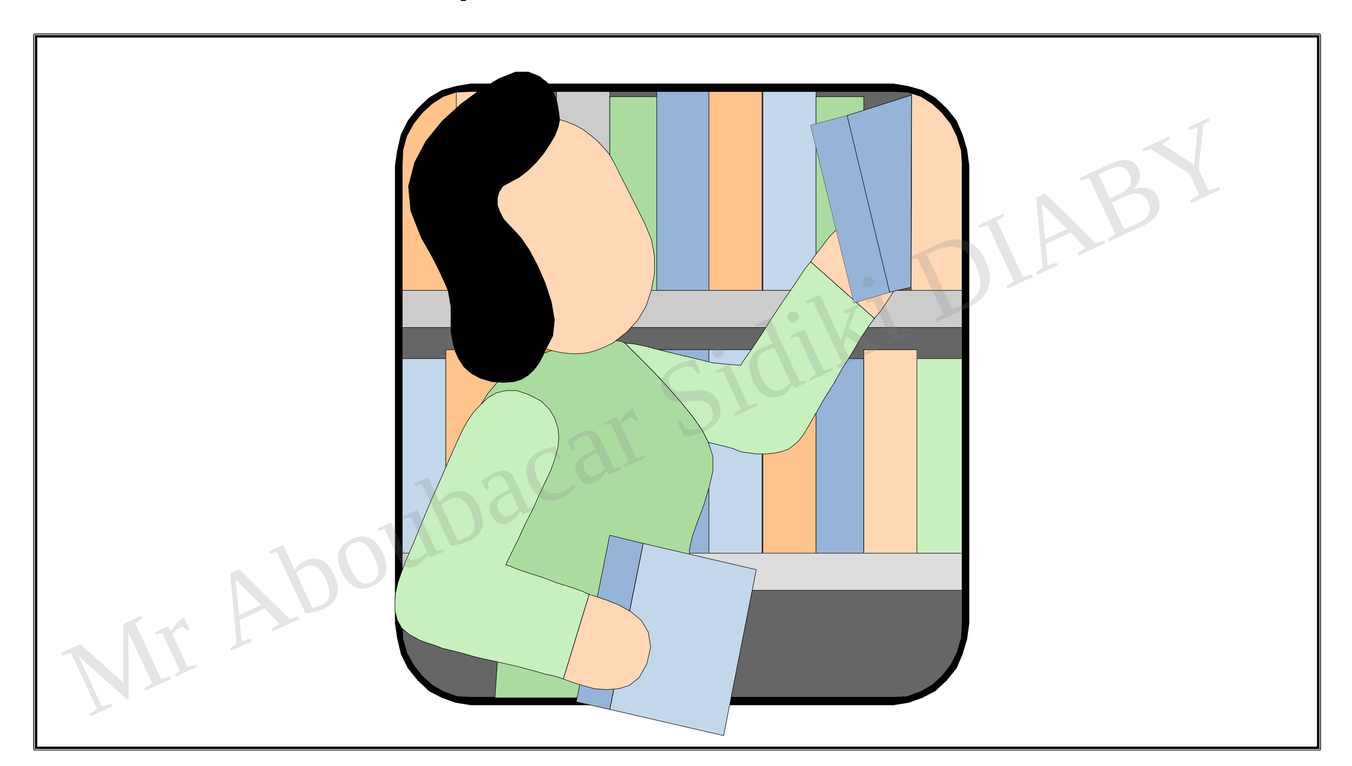
Marche à suivre

1. Programmez la mise à un des LED du simulateur dans l'OB 100 (voir ci-dessus) et la mise à zéro dans le bloc « FC_En_Service » (FC15).
2. Chargez tous les blocs dans la CPU et contrôlez le fonctionnement du programme.
3. Enregistrez votre projet.

Remarques pour la solution à l'exercice

Le bloc de démarrage OB 100 n'est exécuté qu'une seule fois aussi bien en **démarrage manuel** (CPU STOP –> RUN) qu'en **démarrage automatique** (rétablissement de la tension réseau). Au niveau du programme, l'évaluation des informations système de l'OB 100 permet de déterminer le type de démarrage. Les informations système fournies dans les OB sont détaillées dans l'aide en ligne.

13.7. Information complémentaire



13.8. Traitement global des erreurs avec les OB d'erreurs asynchrones

Type d'erreur	Exemple	OB d'erreur	Priorité
Erreur de temps : Temps de cycle maxi. admissible légèrement dépassé Réaction syst. avec OB : RUN sans OB: STOP Temps de cycle maxi. admissible dépassé de plus du double. Réaction syst. avec OB : STOP	Dépassement du temps de cycle maxi. admissible, Appel temporisé d'un OB d'erreur de temps	OB 80	**S7-1500:** paramétrable : 2.. 26
Alarme de diagnostic Réaction syst. sans OB : RUN	Rupture de fil sur un module doté d'une fonction de diagnostic, erreur d'alimentation	OB 82	**S7-/1500: param. 2..26**
Alarme enfichage/débrochage Réaction syst. sans OB : RUN	Débrochage / enfichage d'un module.	OB 83	**S7-/1500: param. 2..26**
Défaillance châssis Réaction syst. sans OB : RUN	Défaut d'un esclave DP ou d'un périphérique IO	OB 86	**S7-/1500: param. 2..26**

Chaque événement donne lieu à une entrée dans le tampon de diagnostic.

Erreurs asynchrones

Les erreurs asynchrones apparaissent de manière asynchrone avec le traitement du programme et ne peuvent de ce fait être attribuées à un emplacement défini du programme.

Erreurs de temps

Elles apparaissent lorsque le temps de cycle courant dépasse le temps de surveillance de cycle paramétré dans HW Config.

Alarmes de diagnostic

Elles sont déclenchées en cas de défaut (par ex. rupture de fil) par les modules dotés d'une fonction de diagnostic, comme les modules analogiques.

Alarmes enfichage/débrochage

Elles sont déclenchées lors de l'enfichage ou du débrochage de modules. Lors de l'enfichage du module, le système d'exploitation vérifie si le type de module enfiché est correct. Cette fonction permet de réaliser un enfichage/débrochage en cours d'exploitation.

Défaillance châssis

Ce type d'erreur est détecté en cas de défaillance d'un châssis, d'un sous-réseau ou d'une station de la périphérie décentralisée.

13.8.1. Traitement global des erreurs avec les OB d'erreurs synchrones

Type d'erreur	Exemple	OB	Priorité
S7-1200 Erreur de programmation Erreur d'accès Réaction syst. sans OB : **RUN**	Accès à un DB non disponible Accès direct à un module périphérique inexistant ou défectueux	**Pas d'OB disponible**	./.
Erreur de programmation Réaction syst. sans OB : **STOP** **S7-1500**	Accès à un DB non disponible	**OB 121 (seulement sur S7-1500)**	**S7-1500:** paramétrable : 2 .. 26
Erreur d'accès Réaction syst. sans OB : **RUN**	Accès direct à un module périphérique inexistant ou défectueux	**OB 122 (seulement sur S7-1500)**	

Chaque événement donne lieu à une entrée dans le tampon de diagnostic.

Erreurs synchrones

Les erreurs synchrones apparaissent de manière synchrone avec le traitement du programme et peuvent de ce fait être attribuées à un emplacement défini du programme.

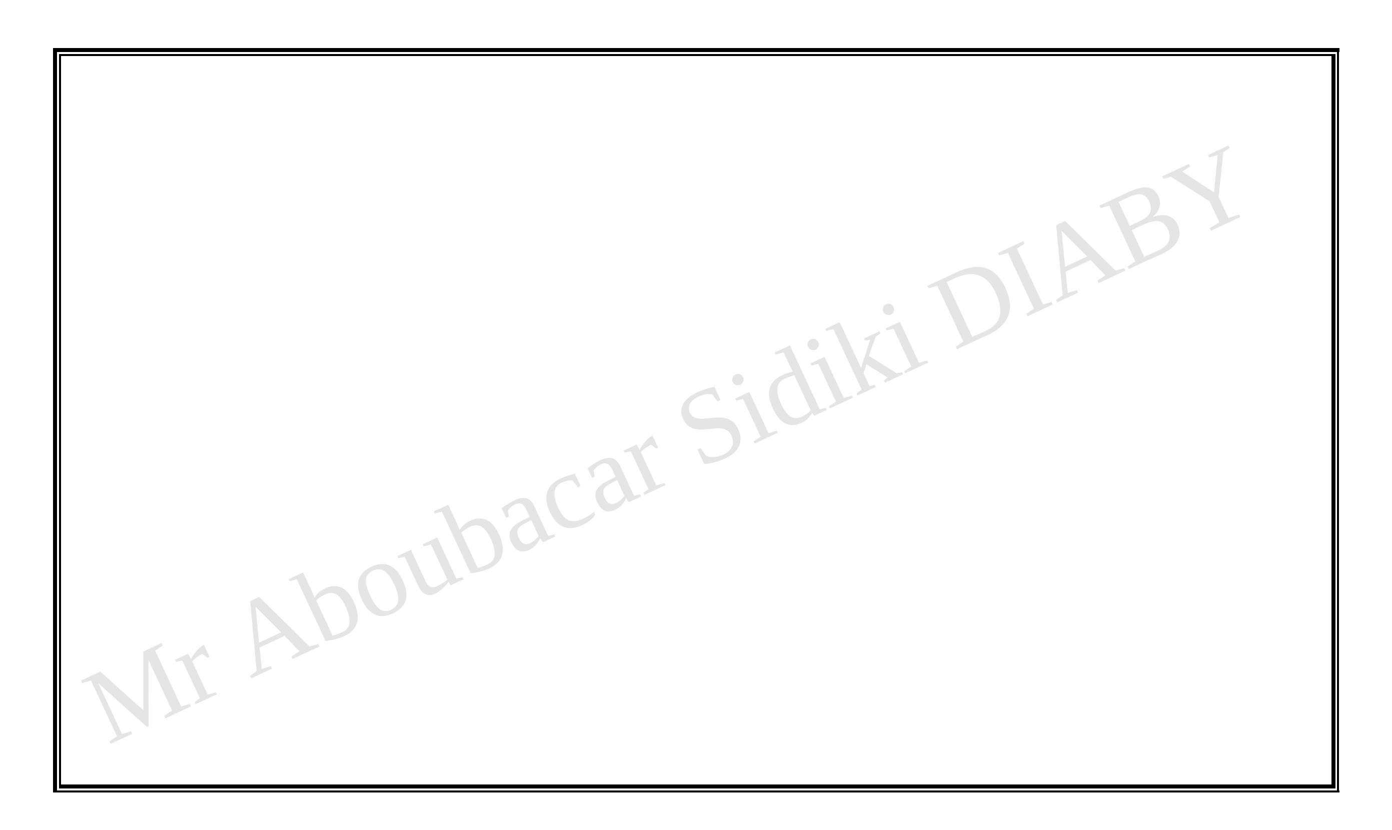
L'OB121 est appelé en cas d'erreur de programmation, l'OB122 en cas d'erreur d'accès. Si l'OB d'erreur synchrone correspondant n'est pas présent dans la CPU en cas d'erreur, la CPU passe à l'état STOP.

S7-1500 :

La priorité des OB d'erreur synchrone est réglable de 22 à 26. Les contenus des registres laissés par le bloc interrompu ne sont pas disponibles dans l'OB d'erreur et ne peuvent pas être manipulés à l'aide de fonctions système.

Mr Aboubacar Sidiki DIABY

13.9. Priorités des OB et réaction du système



Mr Aboubacar Sidiki DIABY

Sommaire

14



14. Périphérie décentralisée

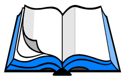
A l'issue du chapitre, vous allez ...



... connaître les systèmes de périphérie décentralisée PROFIBUS et PROFINET

... savoir expliquer le principe de fonctionnement de PROFINET

... Savoir configurer, mettre en réseau et mettre en service une station de périphérie décentralisée PROFINET

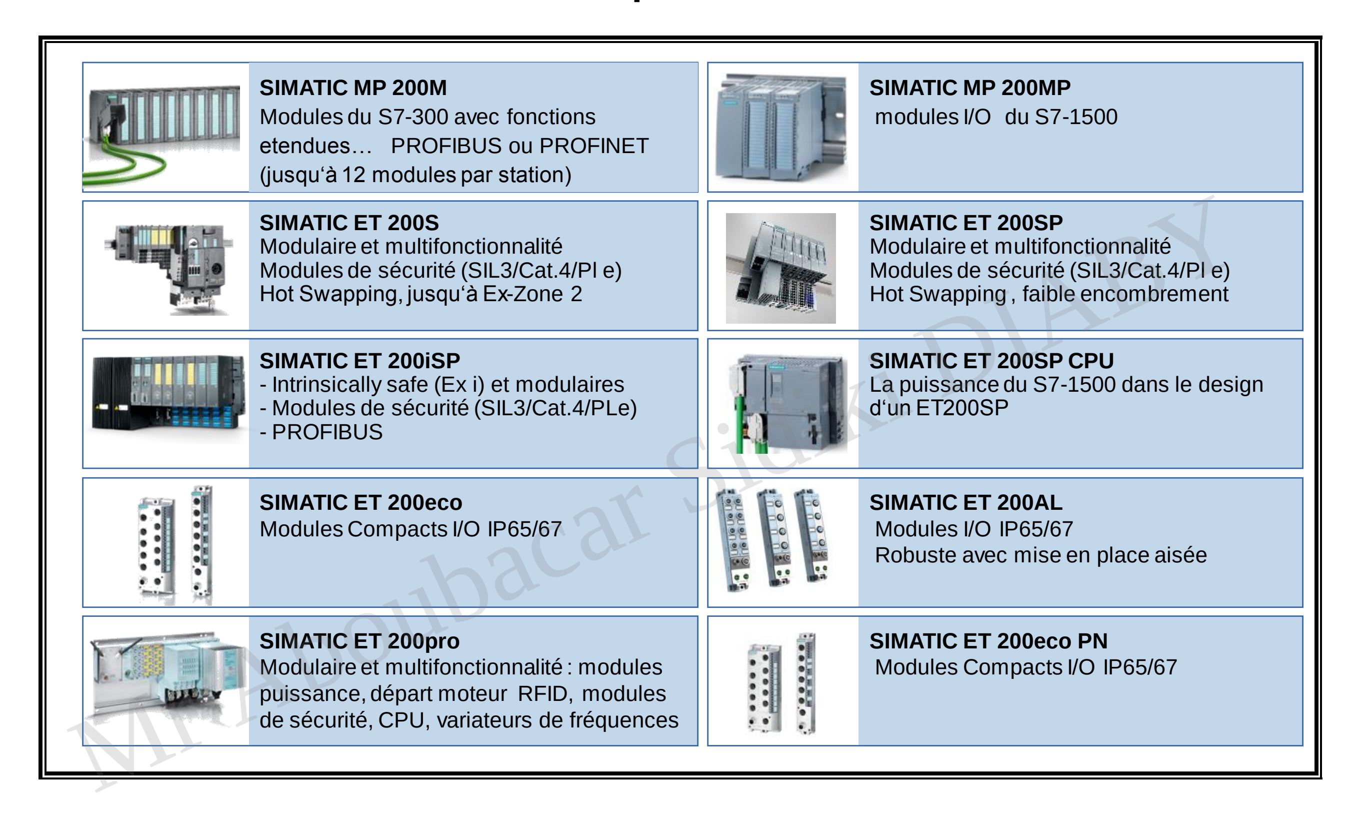


Ce chapitre présente le raccordement des stations périphériques décentralisées.

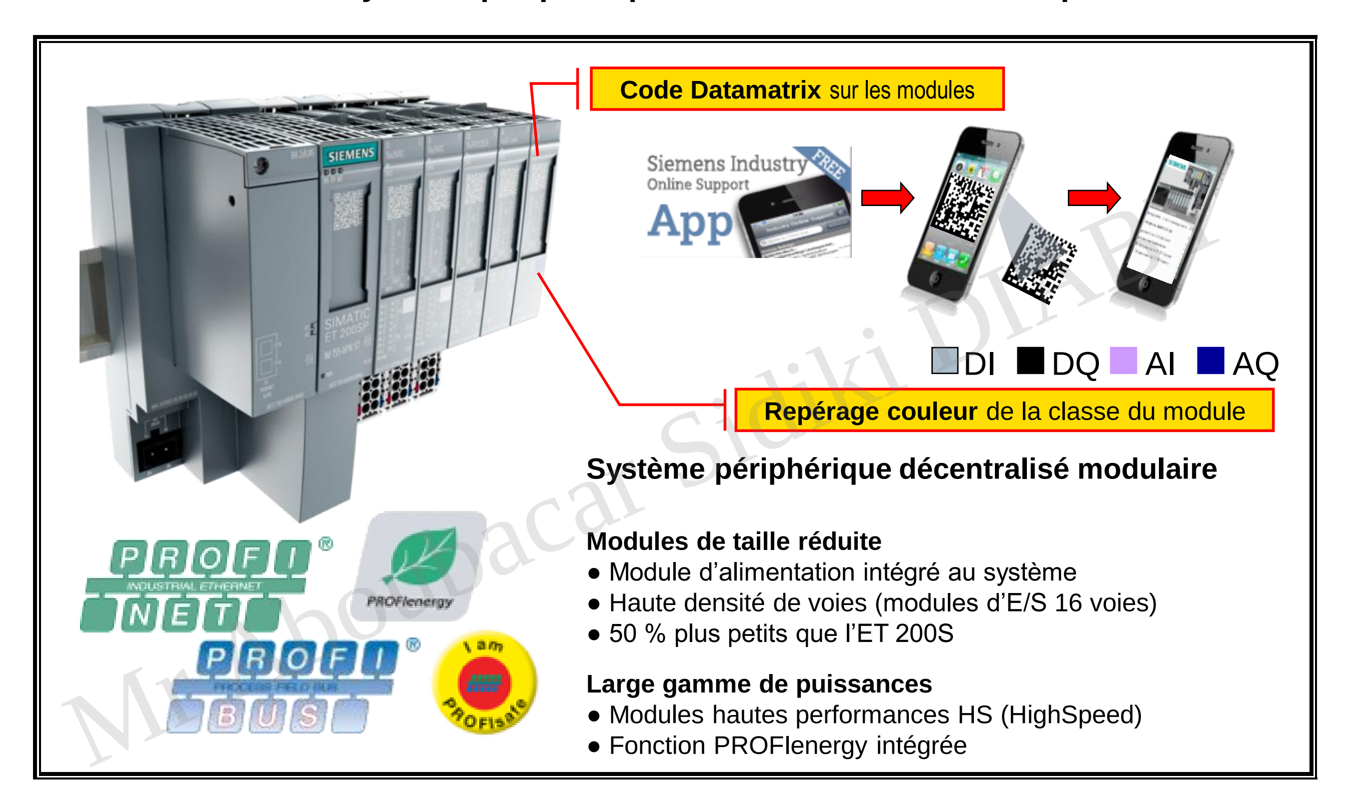
A l'issue de ce chapitre, vous aurez une vue d'ensemble de la configuration des stations périphériques décentralisées.

Objectif pédagogique :
Savoir mettre en service une station périphérique décentralisée.

14.1. Vue d’ensemble de la Périphérie décentralisée



14.1.1. ET 200SP : Système périphérique décentralisé modulaire optimisé en taille



Code Data Matrix (code matriciel 2D)

Ce code est utilisé pour le repérage des modules et peut être notamment photographié et décodé par des smartphones, des PDA, des iPhone, etc.

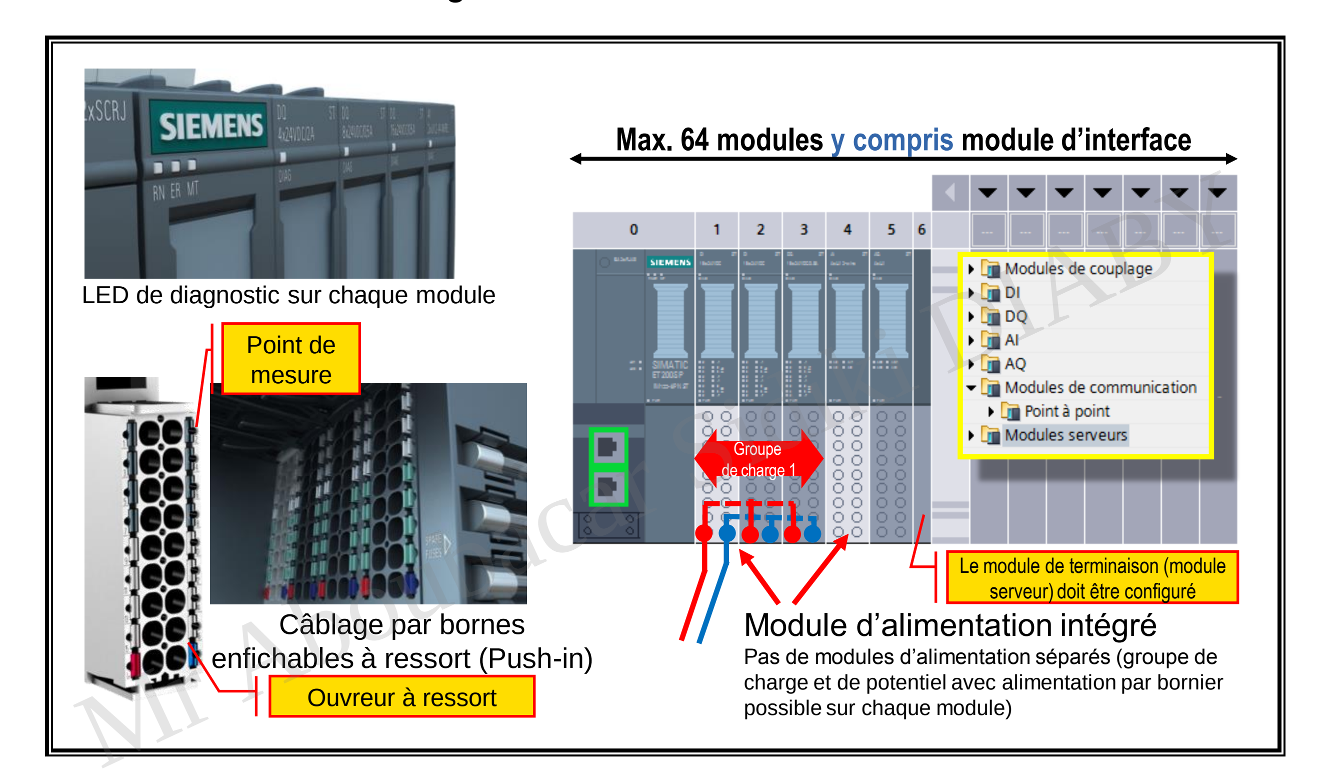
Sur les modules ET200SP, ce code contient un lien vers la page Internet du module correspondant.



Repérage couleur de la classe des modules

Chaque classe de modules est repérée par un carré coloré qui facilite l'identification de la fonctionnalité du module.

14.1.2. ET 200SP : Montage et nombre max. de modules



14.1.3. ET 200MP : Système périphérique décentralisé modulaire



Système périphérique décentralisé modulaire

Modules du système S7-1500 utilisables
- Modules TOR/analogiques, TM, CP/CM, alimentations système (PS)
- Haute densité de voies par ex. 32 DI ou 32 DO par module

Large gamme de puissances
- Coût par voie réduit grâce à la haute densité de voies
- Temps de réaction rapides
- Diagnostic de modules





L'ET200MP permet le raccordement décentralisé de modules périphériques centralisés de la gamme S7-1500.

Le raccordement s'effectue via un module d'interface.

☞ Seul sera livrable dans un premier temps un module d'interface PROFINET.

On dispose d'un fichier GSDML pour STEP 7 V11, STEP 7 V5.5 ou systèmes tiers.

14.1.4. ET 200MP : Montage et nombre max. de modules

Max. 32 modules
→ 1er module = alimentation système (PS)
→ 2e module = module d'interface
→ 3e – 32e module = max. 30 modules périphériques quelconques du système S7-1500

Formation de groupes de charge à l'aide d'alimentations supplémentaires (analogue au montage centralisé du S7-1500)

Max. 32 modules y compris modules d'alimentation et d'interface

Si l'on dispose d'une alimentation 24 V DC adaptée, celle-ci peut alimenter l'IM en face avant même sans module PS au 1er emplacement.

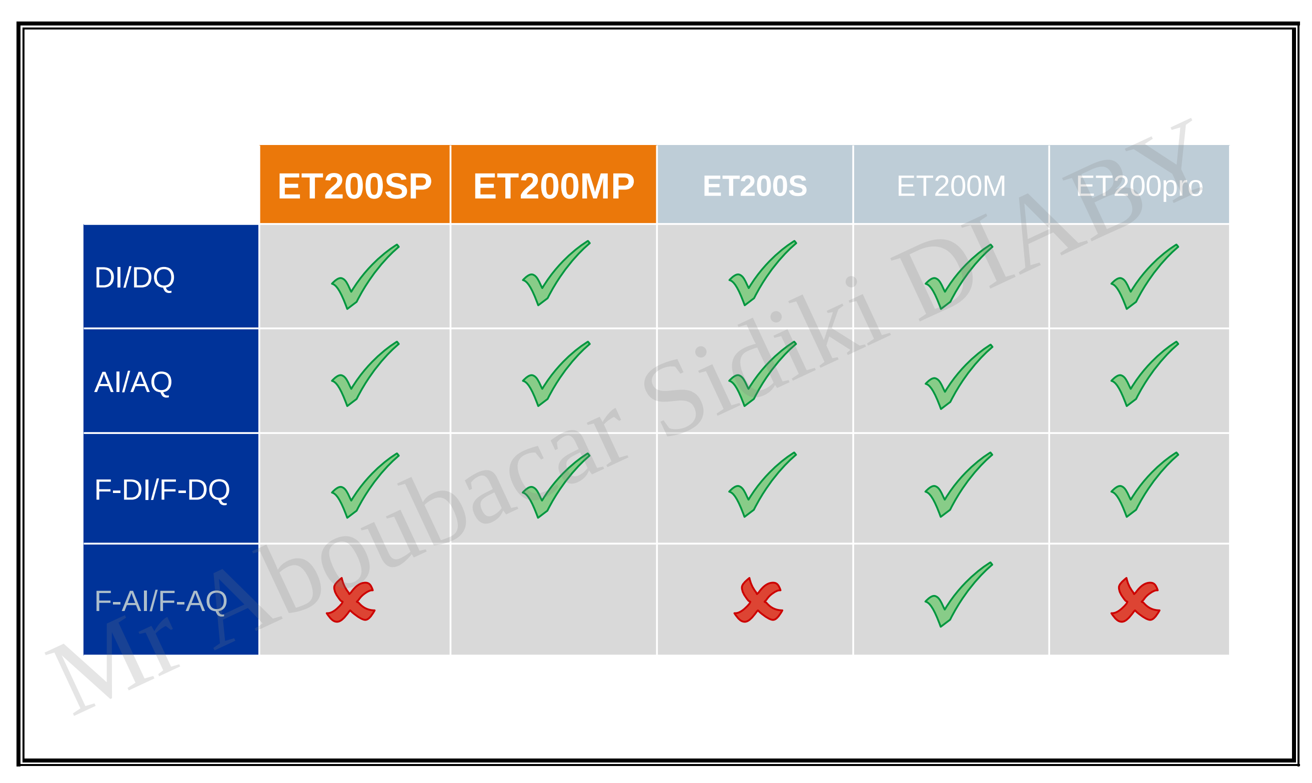
Si les modules périphériques connectés à l'IM155 sont peu nombreux ou peu puissants, ils peuvent être alimentés en tension système par l'IM155 via le bus de fond de panier.

Dans le cas contraire, il convient de réaliser des segments d'alimentation comme avec une configuration avec châssis S7-1500 centralisé.


Module PS **et** 24 V DC en face avant de l'IM155 :

Puissance d'alimentation système du PS
+ Puissance d'alimentation système de l'alimentation intégrée de l'IM155
= Puissance d'alimentation de l'alimentation système dans segment d'alimentation 1

14.1.5. Vue d'ensemble des modules de signaux disponibles (décentralisés)

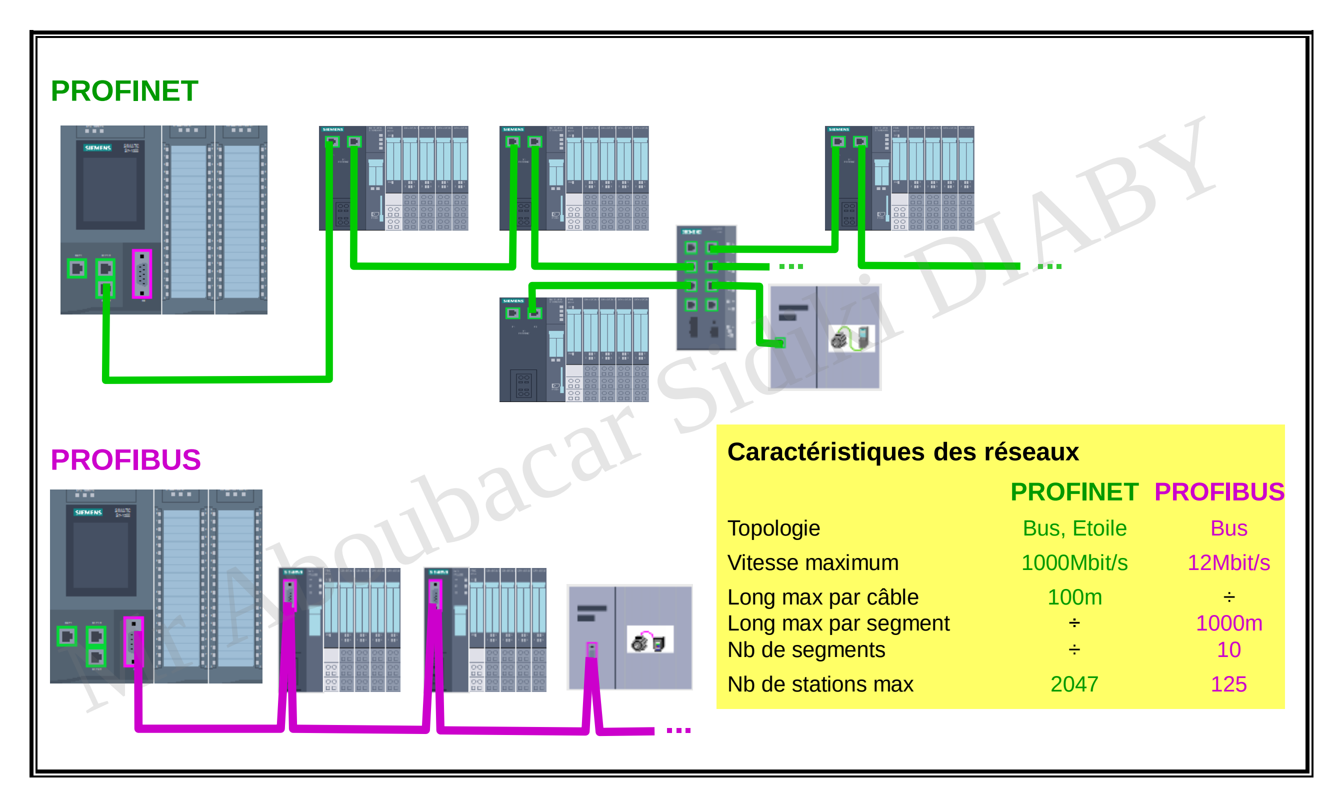


	ET200SP	ET200MP	ET200S	ET200M	ET200pro
DI/DQ	✓	✓	✓	✓	✓
AI/AQ	✓	✓	✓	✓	✓
F-DI/F-DQ	✓	✓	✓	✓	✓
F-AI/F-AQ	✗		✗	✓	✗

Les gammes de produits ET200SP et ET200MP seront étendues au fil des prochaines années de manière à couvrir entièrement les domaines d'application des ET200S et ET200M.

L'ET200pro (module d'interface pour le raccordement à PROFINET ou PROFIBUS) continuera d'être proposé en degré de protection IP65/67 pour une utilisation directe sur la machine.

14.2. Principaux bus de terrain utilisés par le SIMATIC S7 – Vue d'ensemble



Caractéristiques des réseaux

	PROFINET	PROFIBUS
Topologie	Bus, Etoile	Bus
Vitesse maximum	1000Mbit/s	12Mbit/s
Long max par câble	100m	÷
Long max par segment	÷	1000m
Nb de segments	÷	10
Nb de stations max	2047	125

Il existe différents bus de terrain pour le raccordement de la périphérie décentralisée.
Les principaux utilisés par le SIMATIC S7 sont les suivants :

- PROFINET
  Ce standard de communication pour le niveau terrain permet de raccorder des appareils de terrain décentralisés via Industrial Ethernet.
  Conçu pour le milieu industriel, Industrial Ethernet est un réseau de cellule conforme à la norme internationale IEEE802.3 (Ethernet). Il permet de réaliser des solutions de communication performantes et ouvertes caractérisées par une vitesse de transmission élevée.
- PROFIBUS
  Ce système est destiné à la réalisation de réseaux de cellules composés d'un nombre réduit de stations. Grâce à sa conformité aux exigences de la norme EN 50170, il offre une ouverture qui permet de connecter les constituants normalisés de tous les fabricants.

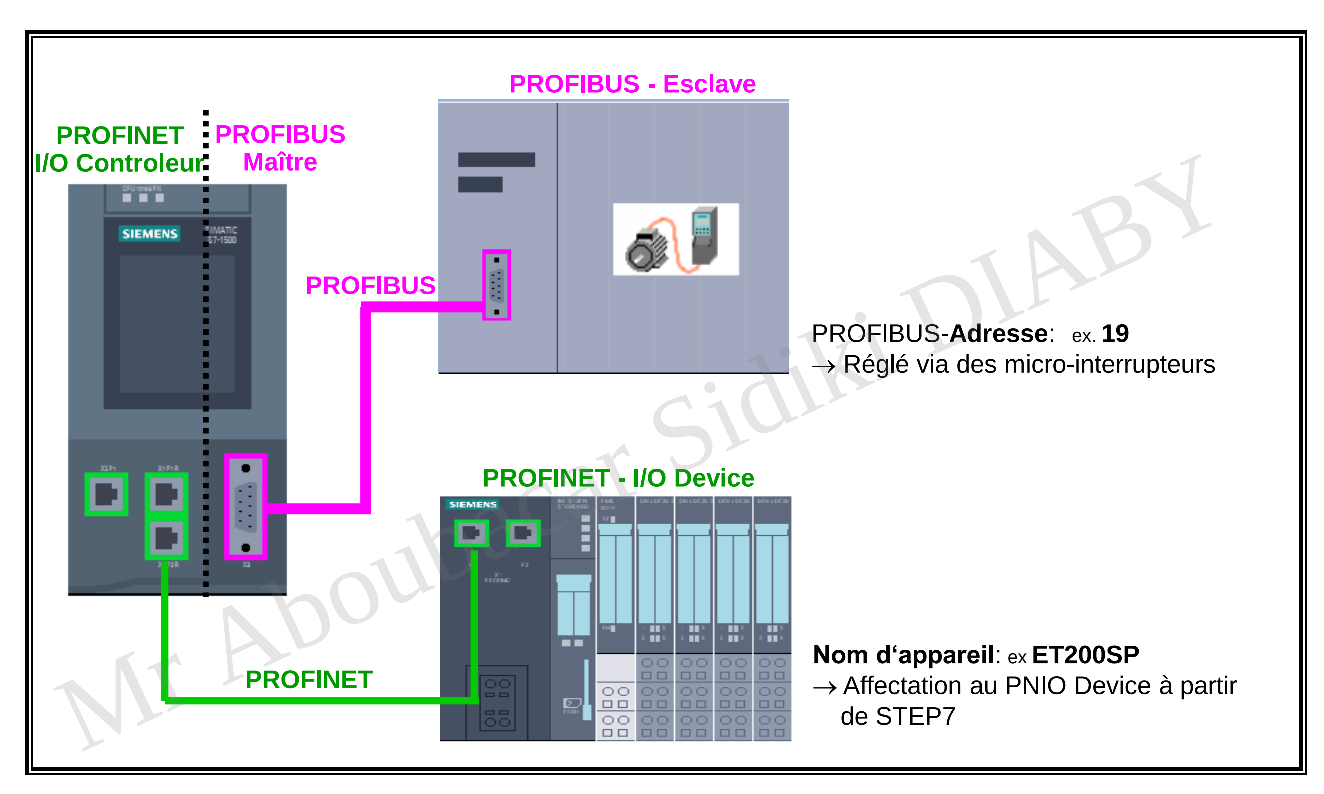
Le choix du bus approprié doit se faire en fonction des exigences de l'application, car ces deux systèmes sont très différents en termes de caractéristiques physiques et de techniques de communication.

Longueur de câble, longueur de segment

Avec PROFIBUS, les signaux des lignes de raccordement des modules doivent être amplifiés au-delà de la longueur maximale admissible de 100 ou 1000 m (selon la vitesse de transmission utilisée).

Avec PROFINET, chaque constituant raccordé prend en charge cette fonction d'amplification. Seule la longueur de câble entre deux modules est donc importante.

14.2.1. Etablissement d’une liaison Contrôleur PNIO↔Périphérique IO ou Maître DP↔Esclave

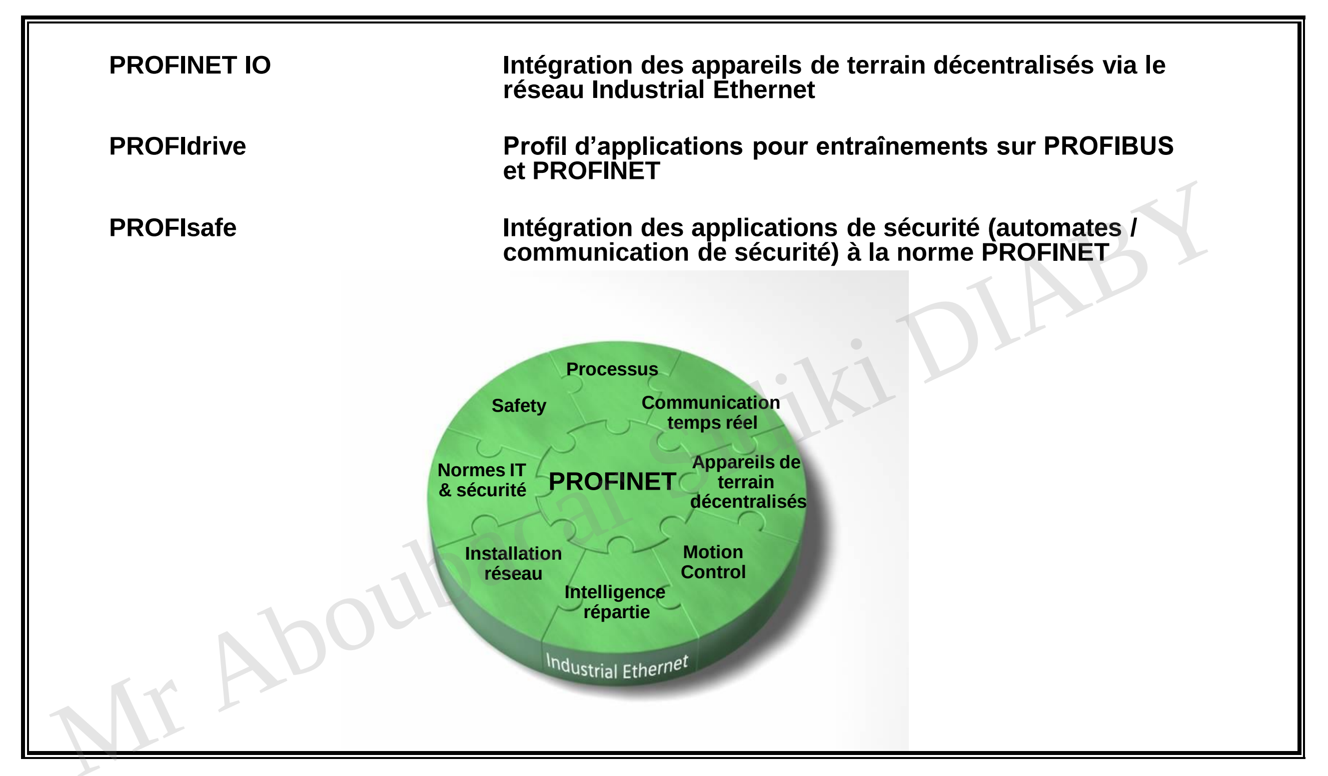


Au démarrage, la CPU recherche les périphériques PNIO (PNIO Devices) ou les esclaves DP configurés et les paramètre selon la configuration matérielle chargée.

Lors de l’identification des modules périphériques, la CPU distingue entre deux systèmes de bus de terrain :

- PROFIBUS
  Les esclaves DP configurés sont recherchés via l’adresse PROFIBUS paramétrée.
  Le paramétrage s’effectue généralement via un commutateur DIP situé sur l’esclave.
- PROFINET
  Le périphérique PNIO (PNIO Device) configuré est recherché via le nom d’appareil qui lui a été affecté. L’affectation du nom d’appareil s’effectue via une fonction en ligne à partir de l’ingénierie STEP7. L’adresse IP paramétrée est ensuite affectée au périphérique PNIO (PNIO Device) par le contrôleur PNIO (CPU).

14.2.2. PROFINET



PROFINET couvre l'ensemble des exigences de l'automatisation industrielle. Ce standard bénéficie de l'expérience de PROFIBUS et d'Industrial Ethernet. Il a été dès l'origine défini en fonction de critères tels que l'ouverture, la simplicité de mise en œuvre et la possibilité d'intégration de parties existantes de l'installation.

PROFINET est aujourd'hui intégré à la norme IEC 61158.

PROFINET IO

PROFINET IO permet l'intégration directe d'appareils de terrain à Ethernet. Il a été conçu selon un modèle fournisseur-consommateur inspiré du procédé maître-esclave bien connu de PROFIBUS DP. En matière de communication, tous les appareils connectés à Ethernet disposent des mêmes droits. Les appareils de terrain sont toutefois affectés à un automate programmable central lors de la configuration. PROFINET reprend la vision classique de PROFIBUS : le périphérique décentralisé lit les signaux de la périphérie et les transmet à l'automate. Ce dernier les traite et retransmet les informations de sortie au périphérique décentralisé.

PROFIdrive

PROFIdrive permettra à l'avenir de réaliser des commandes d'entraînement très rapides et isosynchrones pour les applications de contrôle de mouvement (Motion Control) hautes performances.

PROFIsafe

Grâce à PROFIsafe, les infrastructures de communication standard existantes peuvent être utilisées pour transmettre simultanément, sur un même câble de bus, les données standard et les données de sécurité. PROFIsafe fait appel aux protocoles de bus existants comme PROFIBUS et PROFINET (appelé « canal noir ») pour transporter les données de sécurité en tant que données supplémentaires (couche PROFIsafe). La communication de sécurité est donc indépendante du système de bus et des réseaux subordonnés.

14.2.3. Types d'appareils PROFINET IO

Contrôleurs PROFINET IO
S7-400
S7-1200/1500
Superviseur PROFINET IO
PG
Commutateur Ethernet
Périphériques PROFINET IO
S7-300
ET200eco
ET200S
Device IO intelligent (i-Device)
Device IO compact
Device IO modulaire

Contrôleur PROFINET IO (PROFINET IO Controller)

Après mise sous tension, le contrôleur PROFINET IO (généralement l’API) établit une liaison logique avec les périphériques IO connectés et paramètre ces derniers (paramètres du module, adresse, etc.).
(Sa fonction correspond à celle d’un maître PROFIBUS de classe 1).

Périphérique PROFINET IO (PROFINET IO Device)

Un périphérique PROFINET IO est un appareil d’entrée/sortie décentralisé connecté via PROFINET IO. (Sa fonction correspond à celle d’un esclave PROFIBUS).

Il existe trois types de périphérique PROFINET IO :

- Périphérique compact IO : configuration fixe
- Périphérique modulaire IO : configuration variable pouvant être étendue ou réduite selon les besoins
- Périphérique intelligent IO : dans ce cas, l’API est configuré non pas comme contrôleur IO mais comme périphérique IO et transmet des données d’E/S à un API de niveau supérieur.

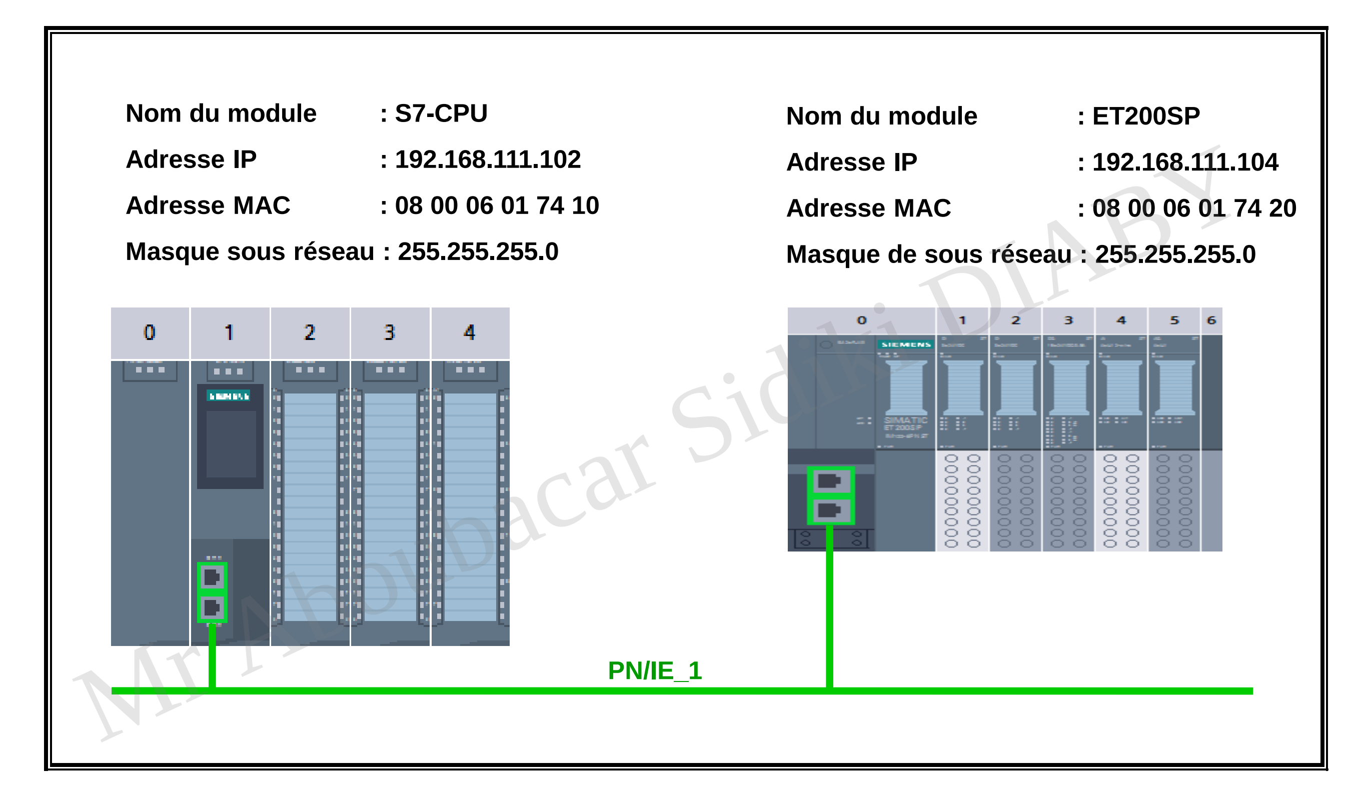
Superviseur PROFINET IO (PROFINET IO Supervisor)

Le superviseur PROFINET IO peut être une console de programmation (PG), un ordinateur personnel (PC) ou une interface homme-machine (HMI) utilisés pour la mise en service et le diagnostic. (Sa fonction correspond à celle d’un maître PROFIBUS de classe 2).

Commutateur Industrial Ethernet (Industrial Ethernet Switch)

PROFINET est basé sur Ethernet. L’utilisation de commutateurs de réseau (switches) est de ce fait toujours nécessaire. Chaque station est reliée à un commutateur via une liaison "point à point", d’où l’appellation "Ethernet commuté". La plupart des appareils PROFINET intègrent d’origine un commutateur 2 ports ou multiport, ce qui simplifie la réalisation des topologies en ligne (comparable à PROFIBUS).

14.2.4. Adresses PROFINET IO



Protocole Internet

Le protocole **IP** (**I**nternet **P**rotocol) constitue la base de tous les réseaux TCP/IP. Il élabore les datagrammes (paquets de données spécialement adaptés au protocole IP) et assure leur transport au sein du sous-réseau local ou leur routage vers d'autres sous-réseaux.

Nom d'appareil PROFINET

Un nom unique et enregistré de manière rémanente dans l'appareil doit être affecté à chaque périphérique IO. Ces noms permettent le remplacement des appareils sans PG/PC.

Adresses IP

Les adresses IP se composent de 4 octets. En notation décimale à point, chaque octet de l'adresse IP est exprimé par un nombre décimal compris entre 0 et 255. Les quatre nombres décimaux sont séparés entre eux par des points.

Masque de sous-réseau

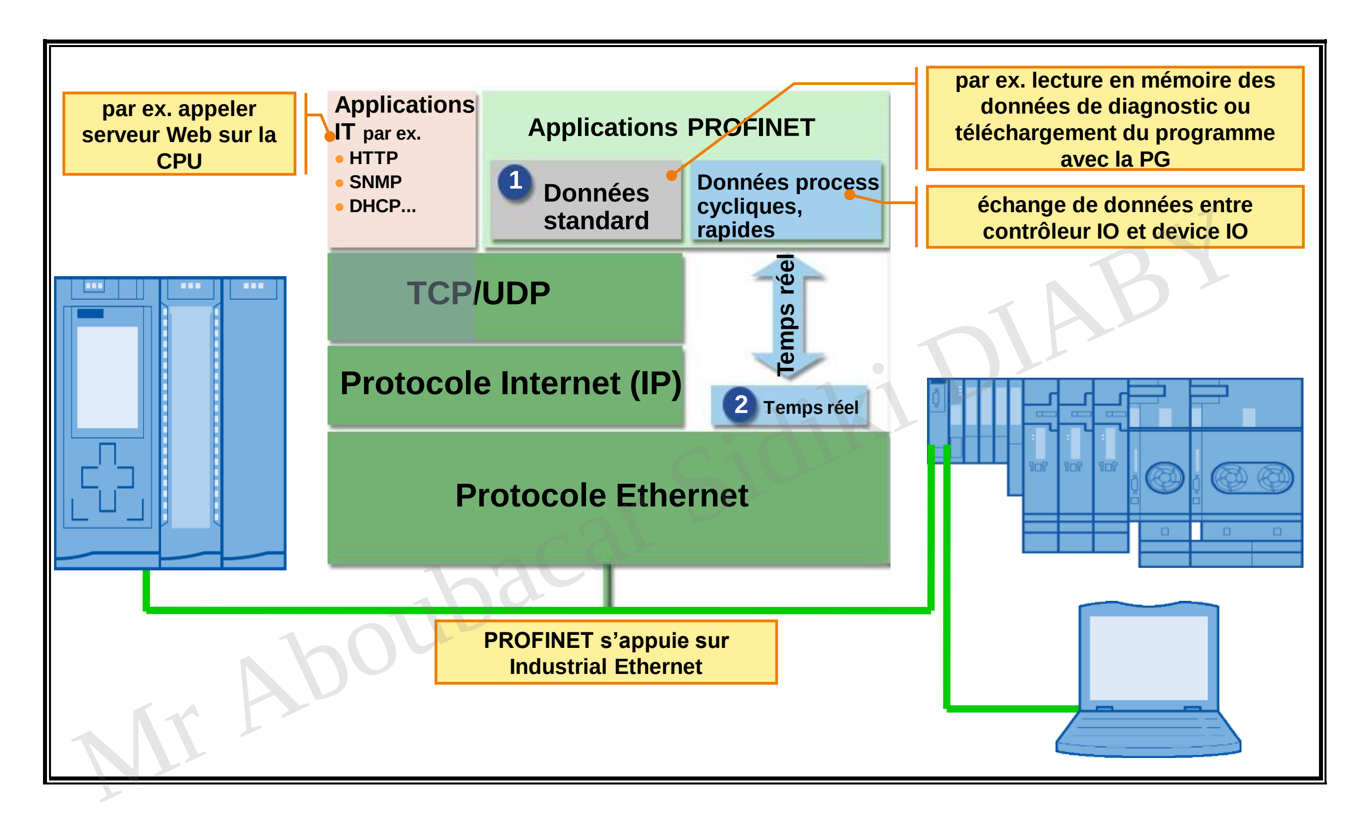
Le masque de sous-réseau permet de subdiviser l'adresse IP en adresse de réseau et adresse d'ordinateur. Il permet en outre de subdiviser un réseau en plusieurs sous-réseaux. Une partie de l'adresse d'ordinateur est alors utilisée comme adresse de sous-réseau. Ces possibilités facilitent l'adaptation des réseaux aux contraintes organisationnelles et physiques.

Adresse MAC

Chaque interface Ethernet a reçu de son constructeur une adresse fixe et unique au niveau mondial. Cette adresse est appelée adresse matérielle ou adresse MAC (Media Access Control). Elle est mémorisée dans la carte réseau et sert à l'identification sans équivoque de l'interface dans un réseau local.

La coopération entre les constructeurs garantit une utilisation unique de chaque adresse au niveau mondial.

14.2.5. Modèle de communication PROFINET



Canal RT (Real Time)

Pour répondre aux impératifs de temps réel des automatismes, un canal de communication temps réel optimisé, dit canal RT (Real Time), a été spécifié dans PROFINET. Il s'appuie sur Ethernet (couche 2).

L'adressage des paquets de données s'effectue dans ce cas non pas via une adresse IP, mais via l'adresse MAC des appareils concernés. Cette solution minimise considérablement les temps de traitement dans la pile de communication et accroît la vitesse d'actualisation des données des automatismes.

Canal IRT (Isochronous Real Time)

Un niveau de communication supplémentaire, le temps réel isochrone (IRT – Isochronous Real Time), a été développé pour répondre à certaines applications à temps critique. Principales caractéristiques :

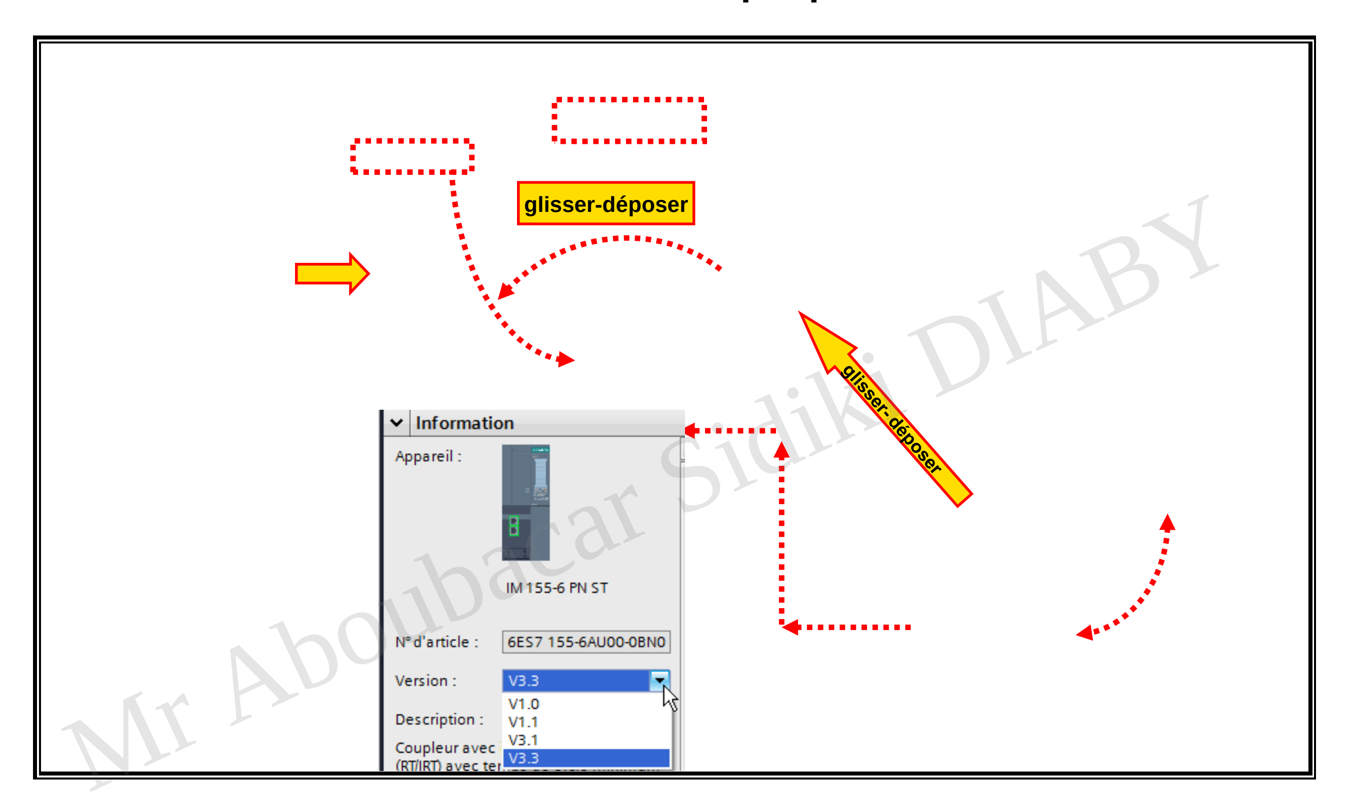
- Transmission de données synchronisées
- Temps de cycle <1ms avec une gigue (incertitude sur les instants de déclenchement) <1µs
- Domaine d'application typique : contrôle de mouvement (Motion Control)

Standards IT

L'intégration du Web dans PROFINET a été principalement axée sur la mise en service et le diagnostic. L'accès aux stations PROFINET à partir d'Internet ou d'Intranet s'effectue à l'aide de protocoles standard comme http. Les données sont transmises dans des formats tels que HTML ou XML et peuvent être visualisées à l'aide de navigateurs standard comme Opera ou Internet Explorer.

Grâce à l'accessibilité mondiale des stations à partir du Web, l'utilisateur peut bénéficier simplement de l'assistance du constructeur lors de la mise en service, du diagnostic, etc. L'accès aux données s'effectue via les serveurs Web intégrés dans les modules.

14.3. Insérer et mettre en réseau une périphérie décentralisée



Insérer une station de périphérie décentralisée

L'insertion d'un module PROFINET IO est réalisée dans la Vue du réseau. Chaque module peut être inséré dans le projet à partir du Catalogue du matériel avec la fonction glisser - déposer.

L'ET200SP inséré n'est pas relié à un Contrôleur et se trouve listé dans la navigation du projet sous « Appareils non affectés » au même niveau que les APIs et les HMIs.

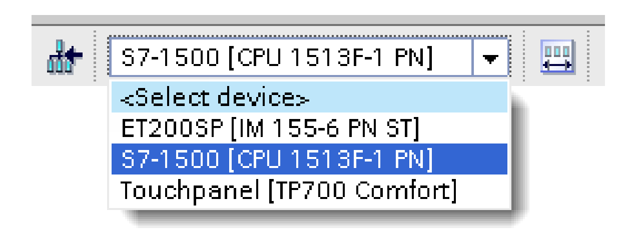
Mettre en réseau la périphérie décentralisée

Après l'insertion de l'IO-Device ET200SP, il s'agit de lui affecter un Contrôleur IO et de le mettre en réseau avec une CPU. Si plusieurs CPU se trouvent sur le réseau, ce n'est que par cette affectation univoque que pourra s'effectuer une coordination et une surveillance des adresses d'E/S du contrôleur IO et de l'IO Device.
Si l'ET200SP est affecté à un contrôleur on le retrouve dans le sous répertoire « Périphérie Décentralisée »

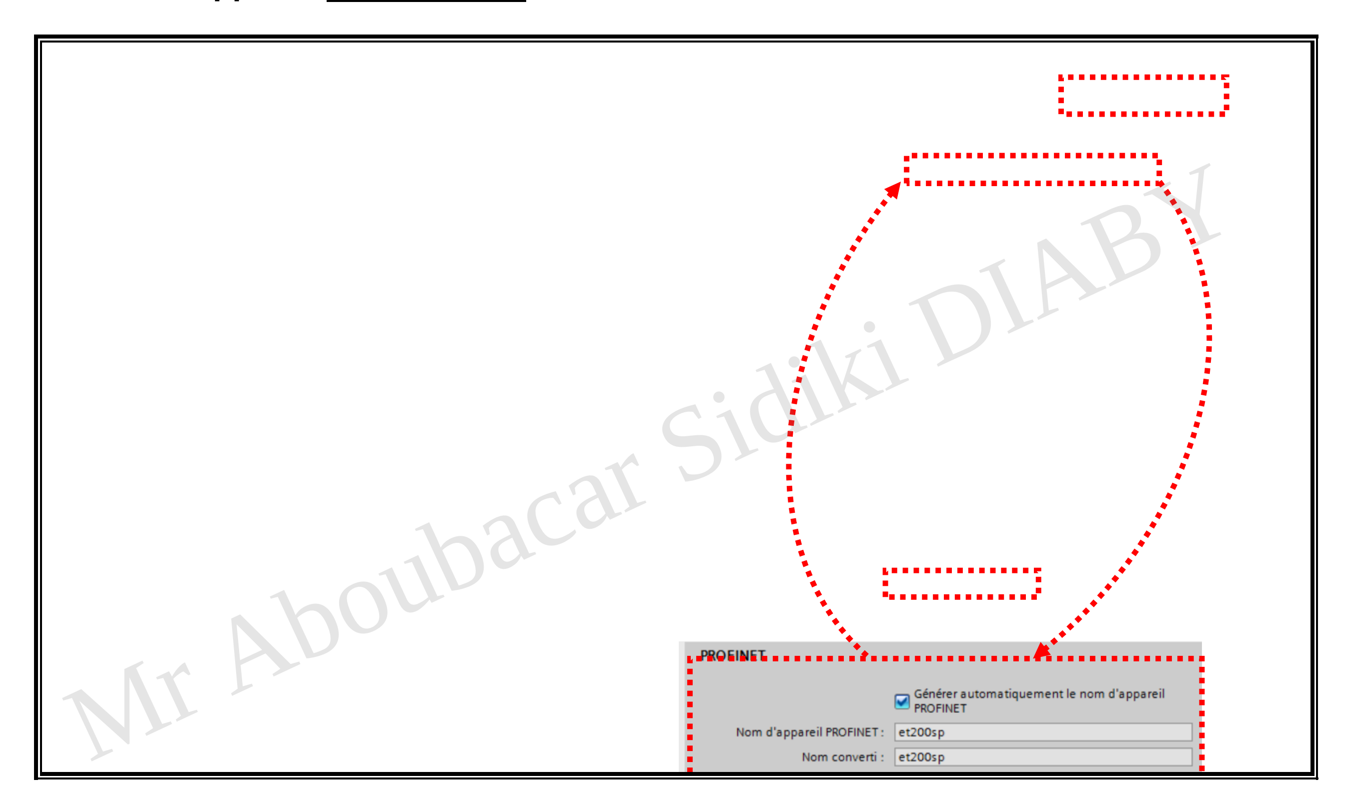
Le module se trouve encore dans le répertoire « Appareils non groupés ».

Modules E/S

Tout comme pour une CPU, vous pouvez configurer et paramétrer individuellement les modules d'entrée et les modules de sortie dans la vue des appareils. Pour cela sélectionnez le module dans la vue des appareils ou via le menu de sélection.



14.3.1. PROFINET IO Device ET200SP: Attribuer une adresse IP et un nom d'appareil HORS LIGNE



Adresse IP, masque de sous réseau et nom d'appareil PROFINET

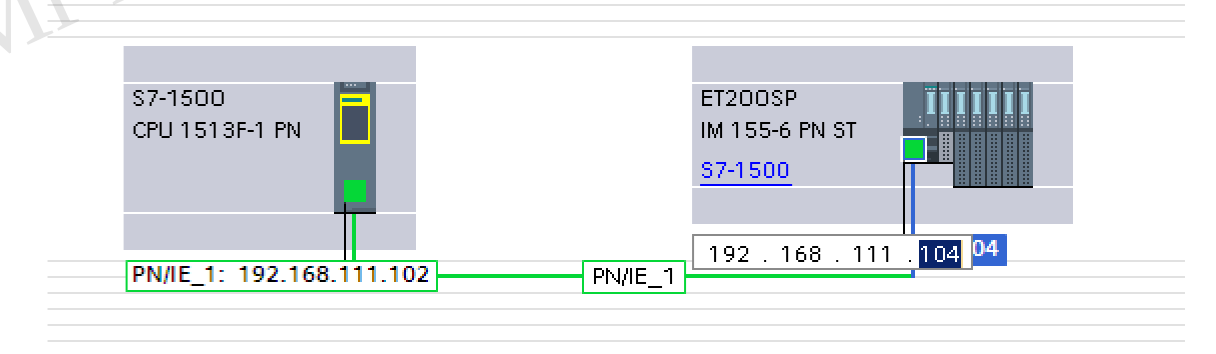
La communication avec le contrôleur IO nécessite une attribution de nom PROFINET Hors Ligne pour l'IO -Device (ET200SP). Le contrôleur IO fourni une adresse IP conforme pour l'IO Device. Si une adresse IP Hors Ligne est attribuée à l'IO-Device , cette adresse sera reprise en tant qu'adresse IP. Ces paramètres sont chargés avec la PG/PC dans le contrôleur IO (CPU), qui les transfère avec d'autres paramètres (comme par ex.les adresses E/S) à l'IO-Device (ET200SP).

Attention:

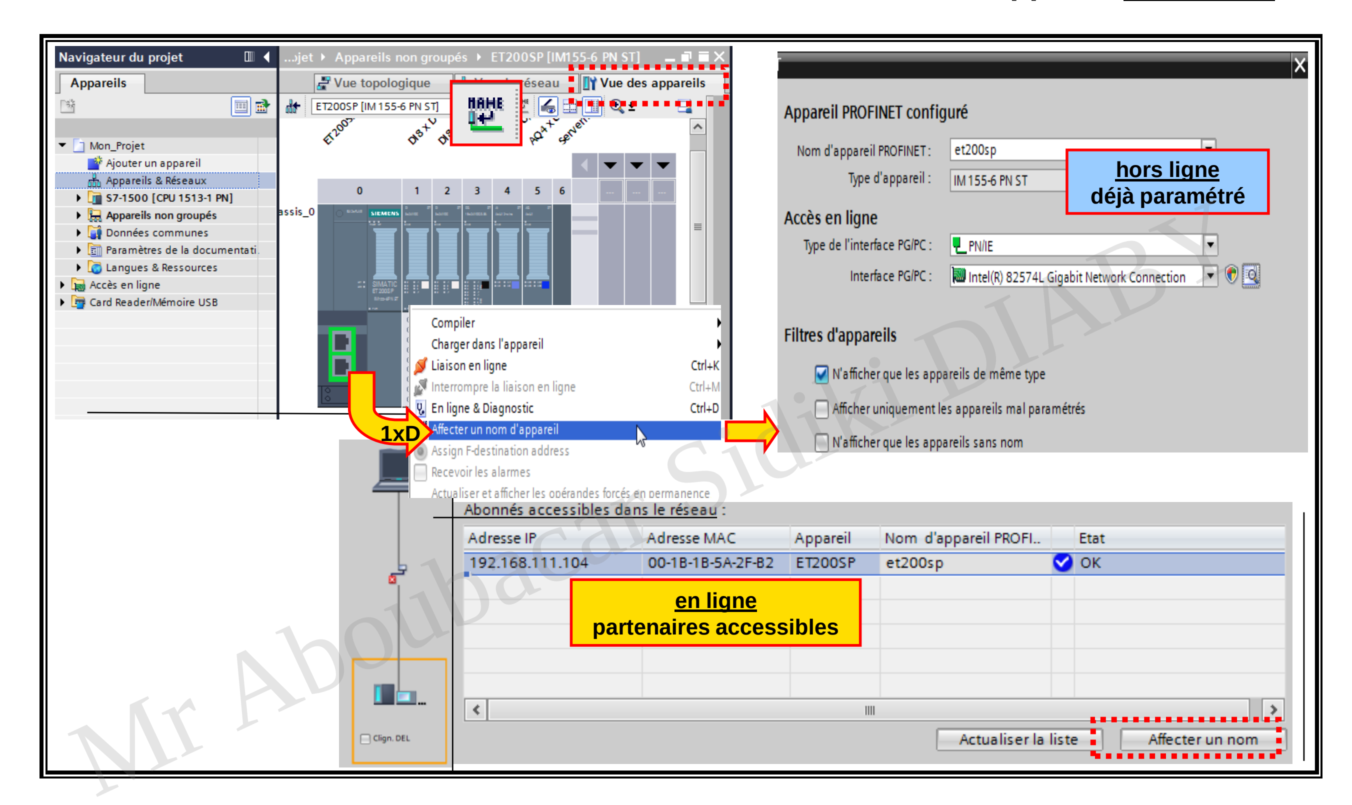
Lors du transfert du paramétrage Hors ligne vers l'appareil En ligne c'est uniquement le nom d'appareil qui est primordial, pas l'adresse IP. Le nom d'appareil PROFINET paramétré Hors ligne et le nom d'appareil PROFINET En Ligne doivent correspondre. Si l'IO-Device a un autre nom d'appareil PROFINET ou n'en dispose pas encore, alors le contrôleur IO ne peut pas transférer la configuration matérielle ou le paramétrage à l'IO-Device, ce qui bloquera le démarrage du système PROFINET.

Remarque :

Les addresses IP peuvent être renseignées directement dans la vue graphique du réseau. Pour visualiser les addresses IP peuvent être visualisées avec le bouton .



14.3.2. PROFINET IO Device ET200SP: attribuer un nom à un appareil EN LIGNE



Adresse IP et nom d'appareil PROFINET

Le nom d'appareil PROFINET paramétré Hors Ligne et celui affecté à l'IO-Device En Ligne doivent correspondre, car le contrôleur IO vérifie d'abord, lors du démarrage, les noms d'appareil des IO-Devices raccordés puis affecte les adresses IP paramétrées. Si un IO-Device n'est pas accessibles via son nom d'appareils paramétré, alors le contrôleur IO ne peut pas établir de liaison avec l'IO-Device.

L'adresse IP de l'IO Device paramétré Hors ligne et celle de l'IO-Device présent En Ligne ne doivent <u>pas</u> forcément correspondre. Pour le chargement de la configuration matérielle seul le nom PROFINET est primordial. Lors de l'opération, si elle est affectée, l'adresse IP présente est écrasée par l'adresse IP paramétrée Hors Ligne.

Possibilités d'affecter un nom En Ligne

En principe il existe deux 2 possibilités d'attribuer un nom d'appareil PROFINET à un IO-Device :

- Variante 1 (sûre, une erreur de saisie est exclue)

  L'attribution du nom d'appareil est réalisée à partir de la Vue de l'appareil de l'IO-Device .

  Vue des appareils de l'IO-Device → clic droit sur le module d'interface (Emplacement 0) → En ligne&Diagnostic →Fonctions → Affecter un nom (vue ci dessus)

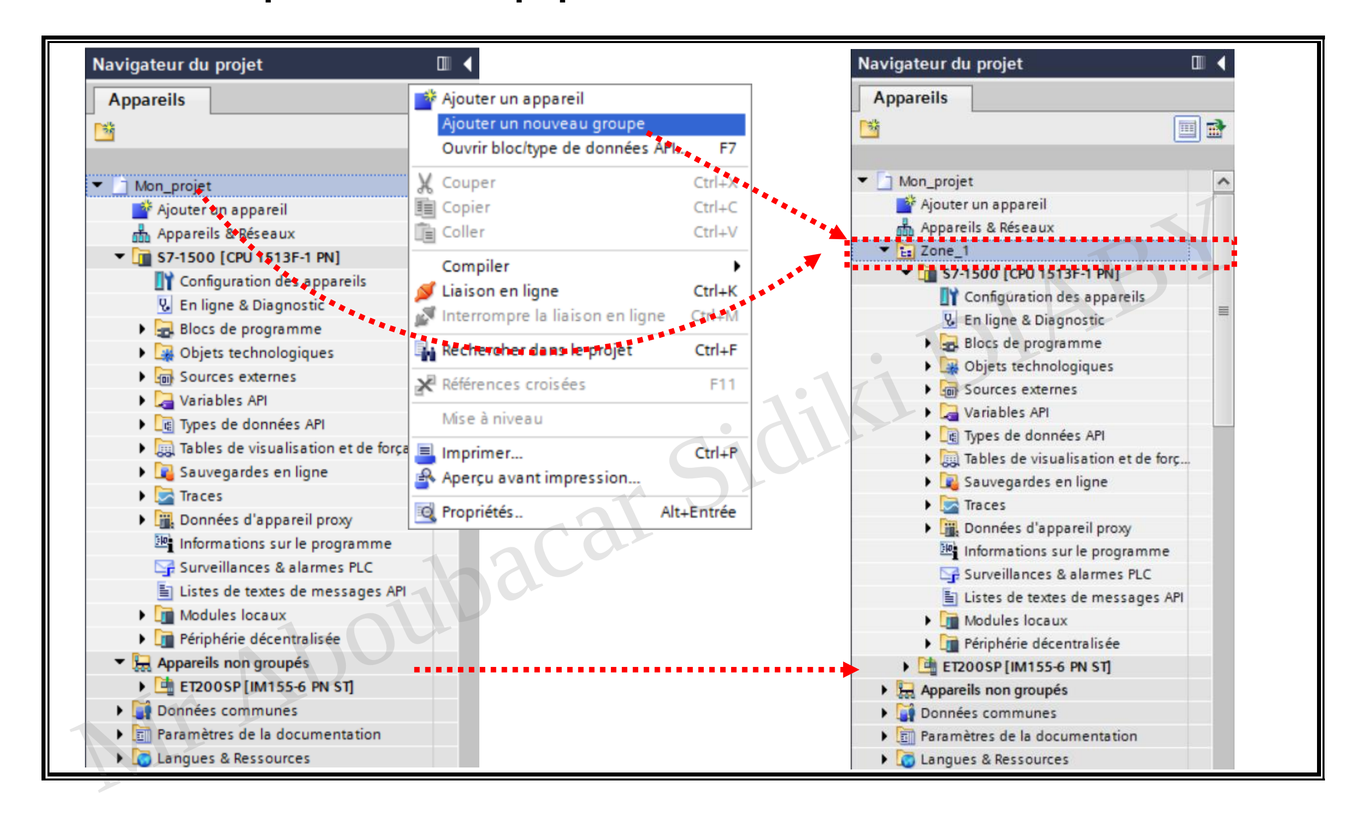
  L'avantage de cette variante est le suivant : le nom de l'appareil Hors Ligne et l'adresse IP sont directement attribués, ce qui exclue une erreur de saisie

- Variante 2 (erreur de saisie possible)

  L'attribution du nom d'appareil et réalisée via « Accès en ligne »:

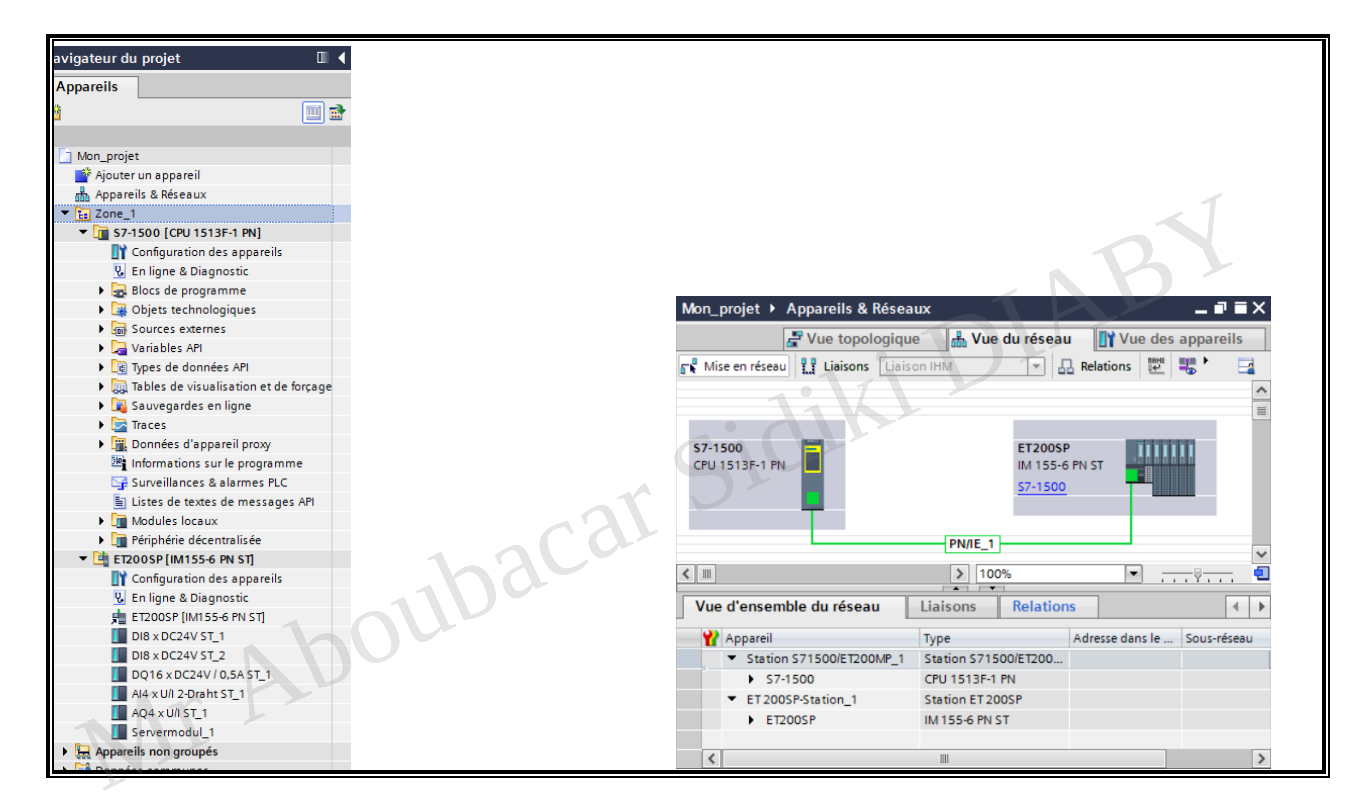
  Vue du projet → Accès en ligne → Interface Ethernet → IO-Device → En Ligne&Diagnostic → Fonctions → Affecter un nom

14.4. Groupement des équipements



Les appareils (périphérie décentralisée également) peuvent tous être insérés directement dans le projet. Pour une meilleure lisibilité du projet, il est recommandé de grouper les modules individuels. Des groupes d’appareils peuvent être créés pour y insérer les appareils par glisser-déposer.

14.5. Enoncé : Mise en service d'une station ET200SP



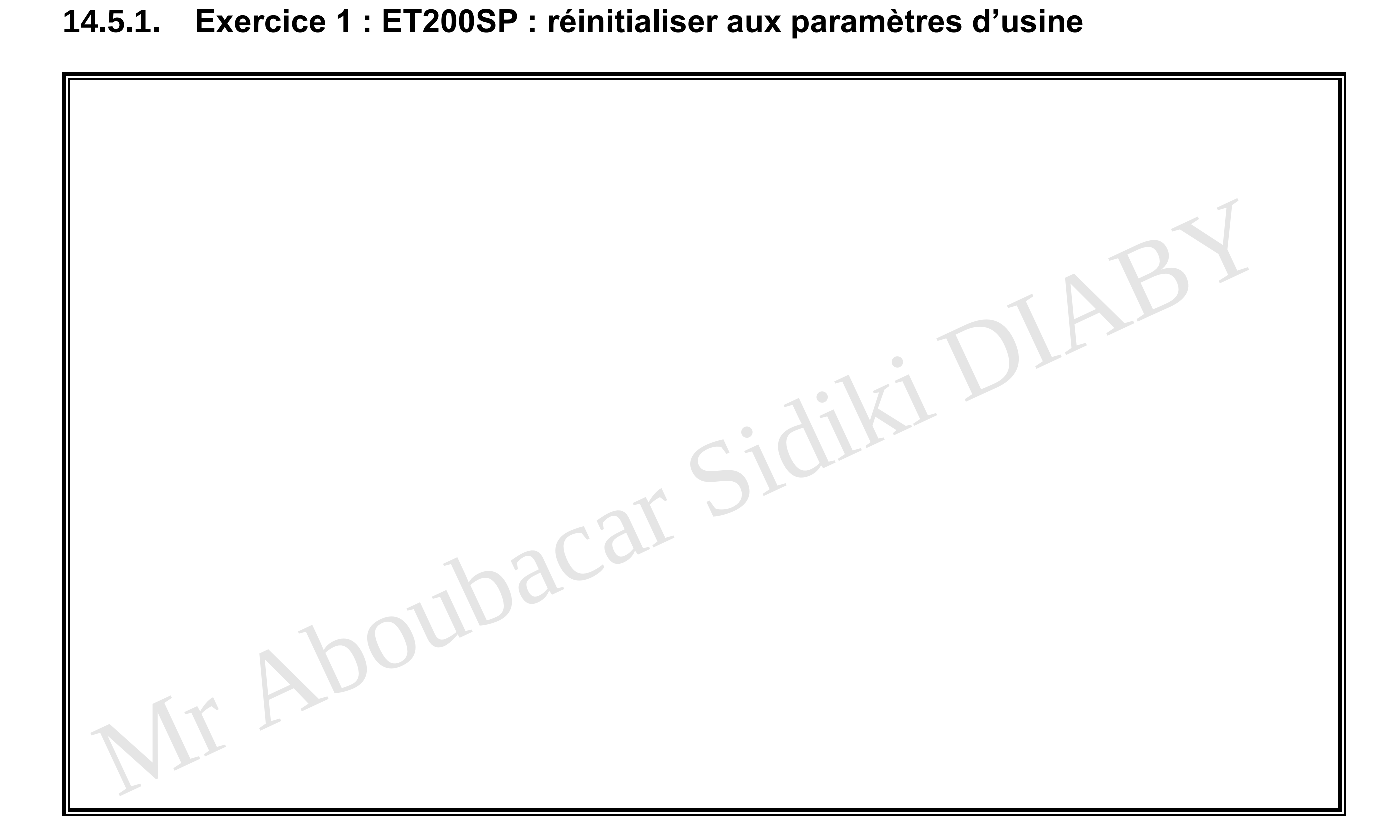
Enoncé

Il s'agit de mettre en service une station de périphérie décentralisée ET200SP, car la bande transporteuse sera pilotée ultérieurement avec ses modules d'entrées et de sorties.

Pour cela, il s'agira de configurer une station ET200SP, de la paramétrer et de la mettre en réseau avec la CPU S7-1500 dans le projet hors ligne.

Après la compilation de la nouvelle configuration matérielle, il faudra la charger dans la CPU qui réalisera, en tant que contrôleur I/O, le paramétrage automatique de l'I/O Device ET200SP.

14.5.1. Exercice 1 : ET200SP : réinitialiser aux paramètres d'usine



Enoncé

Tous les paramètres existants (adresse IP, masque de sous-réseau et nom PROFINET) du module de couplage de l'ET200SP doivent être effacés par une « Réinitialisation aux paramètres d'usine ».

Dans les exercices suivants, vous transmettrez ensuite vos propres paramètres à la station ET200SP.

Mr Aboubacar Sidiki DIABY

Marche à suivre

1. Mettre à jour les abonnés accessibles

Dans le navigateur du projet, ouvrez l'« Accès en ligne », puis sélectionnez l'interface « Intel(R) 82574L Gigabit... ».

Activez ensuite par un double - clic « Mettre à jour les abonnés accessibles » et attendez que la liste soit complétée.

2. Ouvrir « En ligne & Diagnostic » de l'ET200SP

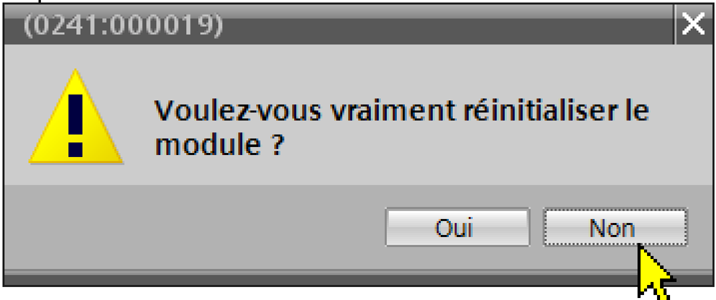
Ouvrez l'ET200SP et activez la fonction « En ligne & Diagnostic » par un double - clic.

Vous aurez peut-être besoin de l'adresse MAC pour l'identification du module. Celle-ci est imprimée sur le boîtier de la tête de station IM155-6 PN.

3. Réinitialiser aux paramètres d'usine

Dans la fenêtre de travail « En ligne & Diagnostic », ouvrez la rubrique « Fonctions », puis sélectionnez « Réinitialiser aux paramètres d'usine ».

Dans la partie droite de la fenêtre de travail, cliquez sur le bouton Réinitialiser.
Répondez « Oui » à la demande de confirmation :



Vérifiez le résultat dans la fenêtre d'inspection sous « Info » + « Général ».

L'appareil a été réinitialisé.

4. Vérifier la réinitialisation aux paramètres d'usine

Vérifiez que la réinitialisation aux paramètres d'usine a bien été effectuée en activant une nouvelle fois « Mettre à jour les abonnés accessibles ».
→ voir Marche à suivre, étape 1.

En tant qu'« abonné », l'ET200SP réinitialisée ne doit à présent s'afficher qu'avec l'adresse MAC.

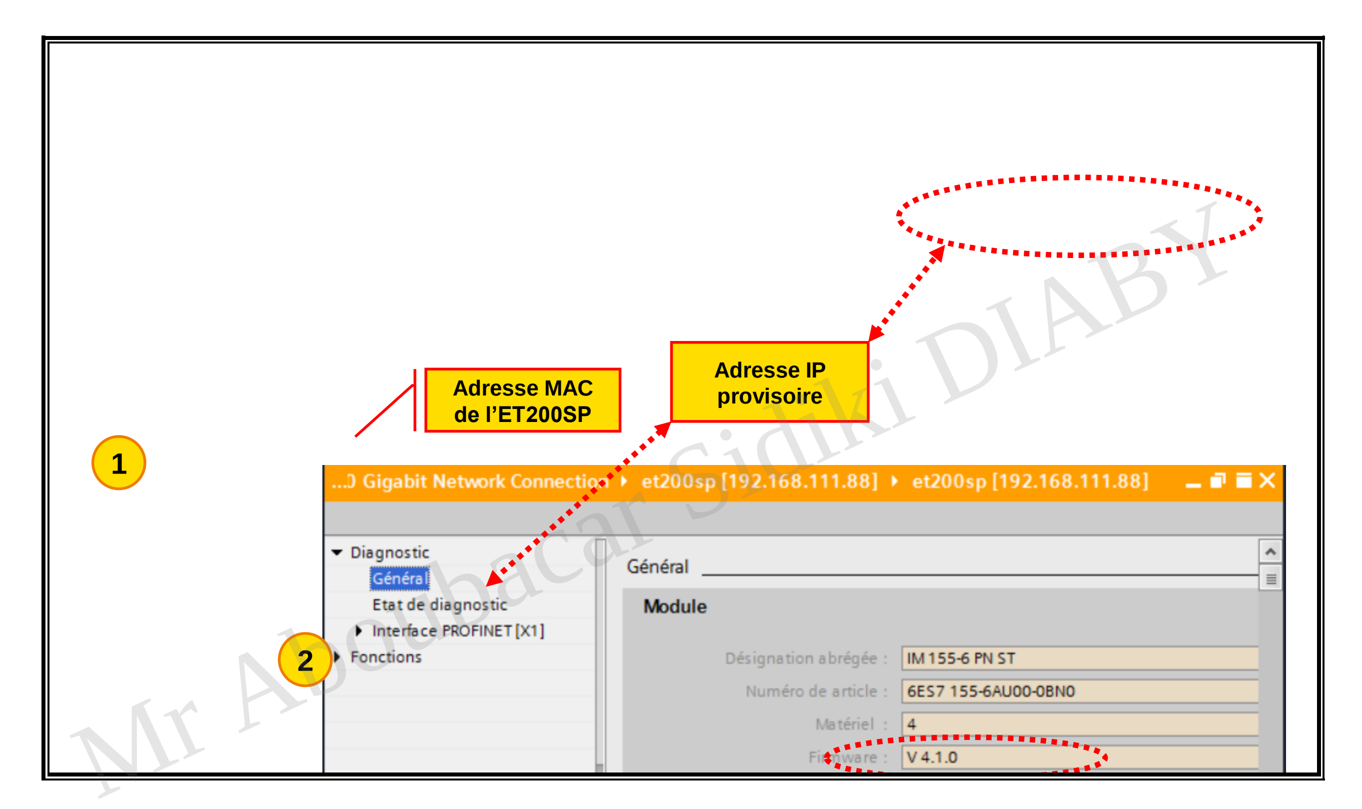
Intel(R) 82574L Gigabit Network Connection
Mettre à jour les abonnés accessibles
Abonné [00-1B-1B-5A-2F-B2]
S7_1500 [192.168.111.102]
touchpanel [192.168.111.101]

Laissez les abonnés accessibles ouverts pour l'exercice suivant.

5. Fermer « En ligne & Diagnostic »

Fermez l'éditeur « En ligne & Diagnostic »."

14.5.2. Exercice 2 : lire la version du firmware de l'ET200SP



Enoncé

Pour pouvoir configurer dans les prochains exercices une ET200SP correspondant exactement au matériel de formation, vous devez connaître sa version de firmware (microprogramme). Ce firmware étant téléchargeable, rien ne vous permet de déterminer avec certitude sa version actuelle. C'est pourquoi vous devez à présent lire en ligne la version courante du firmware de l'ET200SP.

Reportez ensuite la version du firmware déterminée dans la vue d'ensemble de l'exercice 1 du chapitre 1.

Problème

Du fait de la précédente réinitialisation aux paramètres d'usine, l'ET200SP est dépourvue de nom d'appareil PROFINET et d'adresse IP et ne possède plus qu'une adresse MAC (voir figure ci-dessus). Or, l'adresse MAC ne permet pas de lire la version du firmware, ce service de diagnostic exigeant une adresse IP.

Solution

1. Entrez une adresse IP provisoire adaptée au sous-réseau actuel et actuellement non utilisée (par ex. 192.168.111.88).
2. Lisez la version du firmware et notez-la.

Marche à suivre

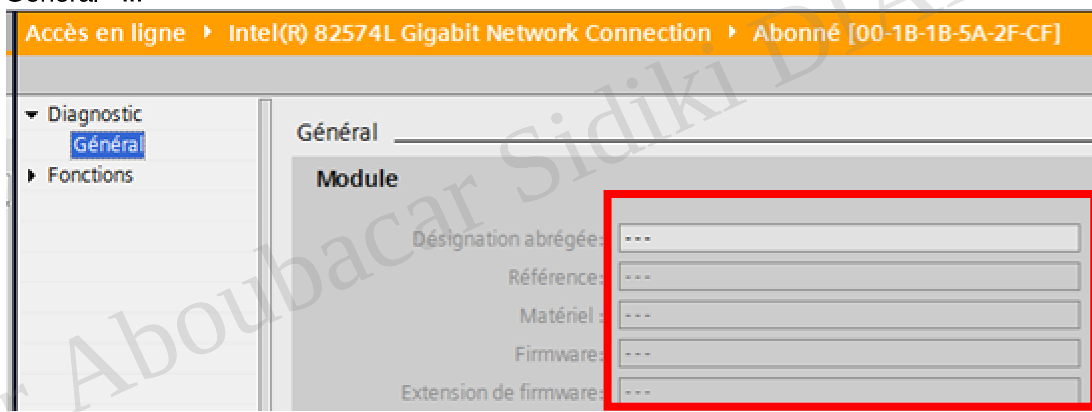
1. Ouvrir « En ligne & Diagnostic » de l'ET200SP

Ouvrez l'ET200SP, puis activez la fonction « En ligne & Diagnostic » par un double-clic.



Si l'exercice 1 a été correctement effectué, l'adresse MAC du module est à présent requise pour l'identification. Celle-ci est imprimée sur le boîtier de la station de tête IM155-6 PN.

Aucune information sur le module ET200SP ne doit s'afficher sous l'onglet « Diagnostic -> Général »...



...sinon, refaites l'exercice 1, car le module n'a pas été réinitialisé.

2. Attribuer une adresse IP provisoire à l'ET200SP

Dans la fenêtre de travail « En ligne & Diagnostic », ouvrez la rubrique « Fonctions », puis sélectionnez « Affectez l'adresse IP ».

Entrez l'adresse IP provisoire indiquée sur la figure ainsi que le masque de sous-réseau et lancez le transfert de l'affectation par [Affecter l'adresse IP].

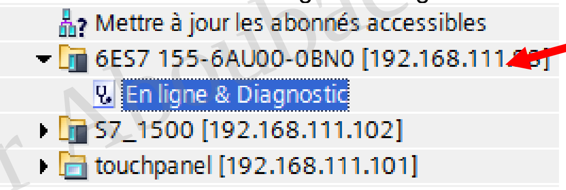
Vérifiez le résultat dans la fenêtre d'inspection sous « Info » + « Général ».

 Les paramètres ont été transférés avec succès.

4. Lire la version du firmware

Dans le navigateur du projet, actualisez à nouveau la liste en cliquant sur « Mettre à jour les abonnés accessibles ».

Dans la liste des abonnés, l'ET200SP s'affiche à présent avec sa référence et son adresse IP. Activez à nouveau « En ligne & Diagnostic ».



Notez la version du firmware affichée sous l'onglet « Diagnostic -> Général ».

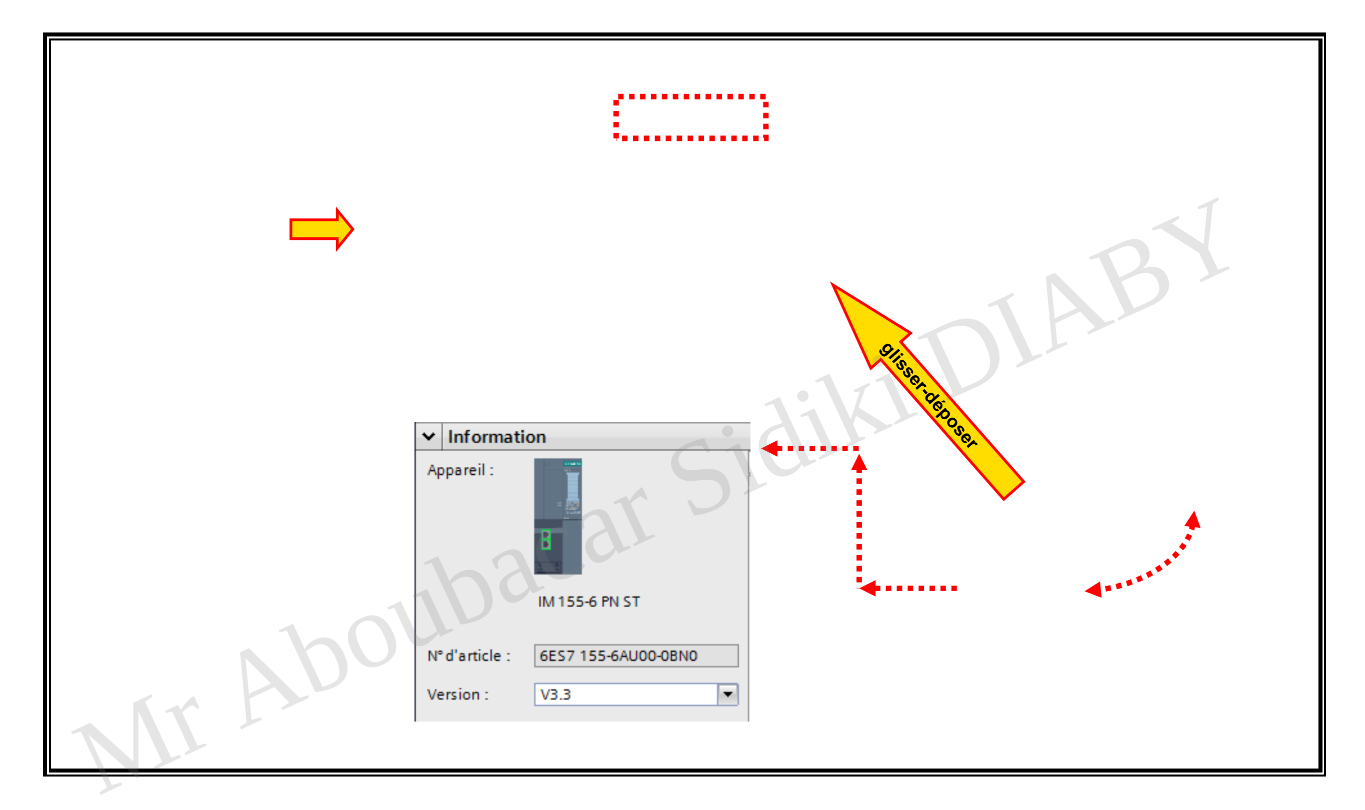
5. Fermer « En ligne & Diagnostic »

Fermez l'éditeur « En ligne & Diagnostic »

6. Fermer « Accès en ligne »

Fermez « Accès en ligne » dans le navigateur du projet.

14.5.3. Exercice 3 : ajouter une station ET200SP



Enoncé

Une station ET200SP doit être ajoutée au projet en tant que station périphérique décentralisée.

L’ajout d’un périphérique PROFINET IO s’effectue dans la « Vue du réseau ». Les différents appareils peuvent être insérés ici dans le projet par simple glisser-déposer à partir du Catalogue du matériel.

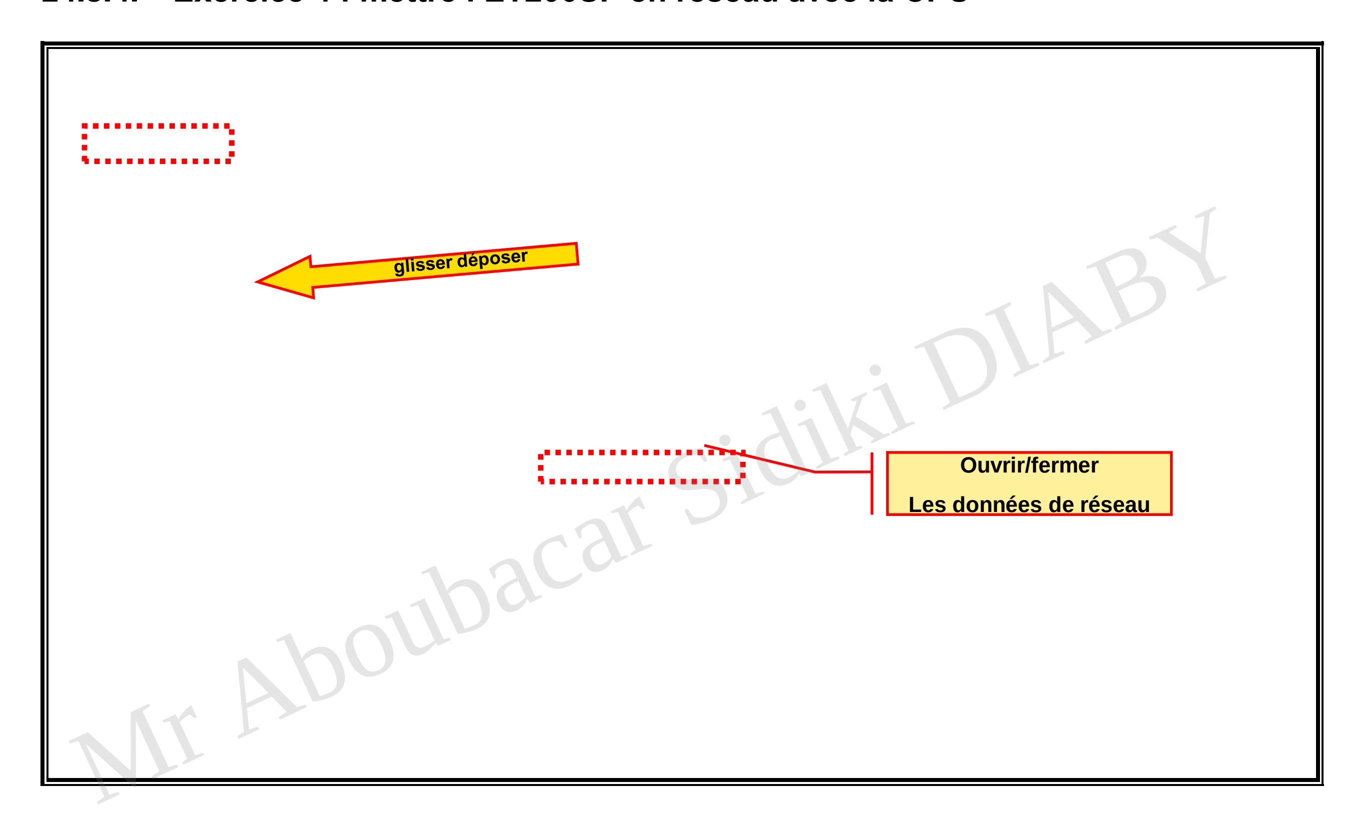
Dans un premier temps, la nouvelle ET200SP ajoutée n’est affectée à aucun automate et apparaît de ce fait, en tant qu’« Appareil non affecté », au même niveau que les CPU et les HMI dans le navigateur du projet.

Mise en œuvre

1. ouvrez par double – clic dans la Vue du projet l’éditeur « Appareils et réseaux »
2. ouvrez la Task - Card du catalogue du matériel puis sous Périphérie décentralisée -> ET200SP -> coupleurs -> PROFINET -> IM155-6PN ST
3. Sélectionnez le module IM utilisé pour la formation, ouvrez la fenêtre « Information « et choisissez la version du firmware du module IM *(dans notre exemple la version v3.3 logicielle est compatible avec la version 4.1 dans la station car elle est inférieure)*
4. Par glisser – poser ajoutez le module IM dans l’éditeur « Appareils et réseaux » (Vue ci dessus)

Veuillez laisser toutes les fenêtres ouvertes, elles seront utilisées dans les exercices suivants!

14.5.4. Exercice 4 : mettre l'ET200SP en réseau avec la CPU



Enoncé

Le périphérique IO ET200SP étant ajouté, vous devez à présent l'affecter à un contrôleur IO ou le mettre en réseau avec une CPU. Si le réseau comporte plusieurs CPU, cette affectation est indispensable pour coordonner ou surveiller les adresses d'E/S des contrôleurs IO et des périphériques IO.

Marche à suivre

1. Sélectionner la mise en réseau

Dans l'éditeur « Appareils & Réseaux », sélectionnez la « Vue du réseau », puis cliquez sur l'onglet « Mise en réseau ».

2. Etablir la liaison entre la CPU et le module

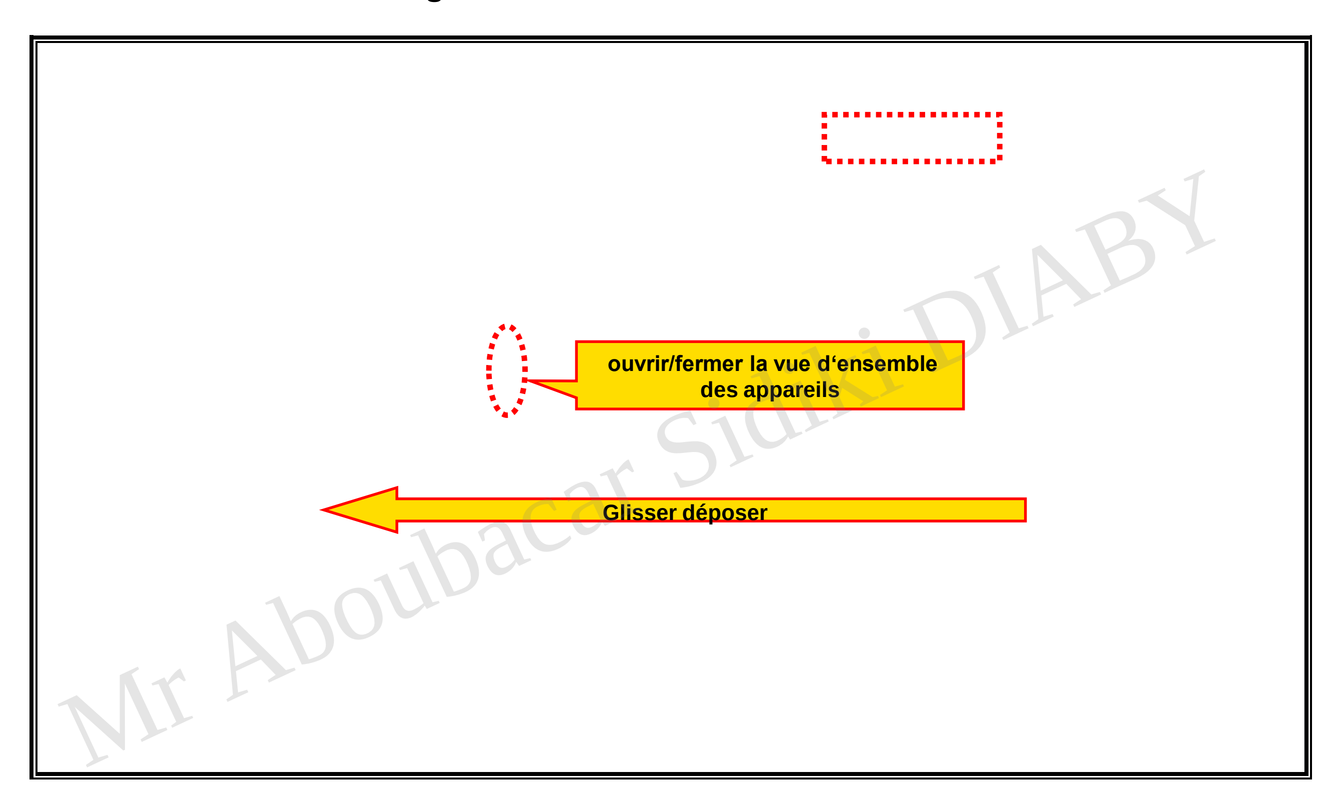
Mettez en réseau l'ET200SP avec la CPU en reliant par glisser-déposer l'interface Ethernet de l'ET200SP avec l'interface Ethernet de la CPU.

☞ Vous pouvez également utiliser le lien Non affectées sur le module.

3. Sélectionnez le nouveau système PROFINET IO

Dans « Communication E/S » vérifiez les partenaires de communication générés.

14.5.5. Exercice 5 : configurer les modules de l'ET200SP



Enoncé

La configuration de l' ET200SP du projet Hors ligne doit être strictement conforme à la configuration du matériel de formation. Portez un soin particulier aux numéros des articles ainsi qu'aux versions des modules.

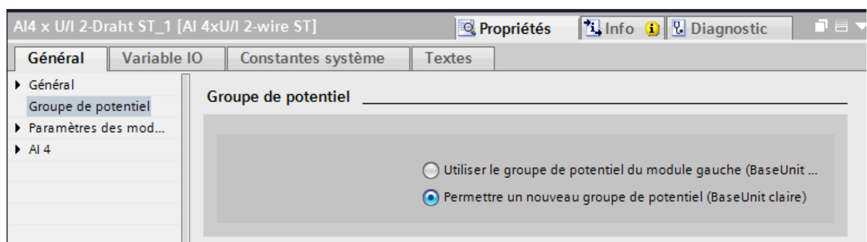
L'ET200SP comporte des modules d'entrées et de sorties TOR et analogiques sur lesquels il faudra, par la suite, raccorder le matériel de formation. Les adresses d'E/S utilisées dans le programme STEP 7 doivent correspondre aux adresses paramétrées pour les modules DI/DQ.

L'adressage actuel se trouve dans la partie inférieure de la fenêtre de travail de l'éditeur « Appareils et Réseaux » dans « Vue des appareils » pour le module. Les adresses sont modifiables dans ce tableau.

Marche à suivre

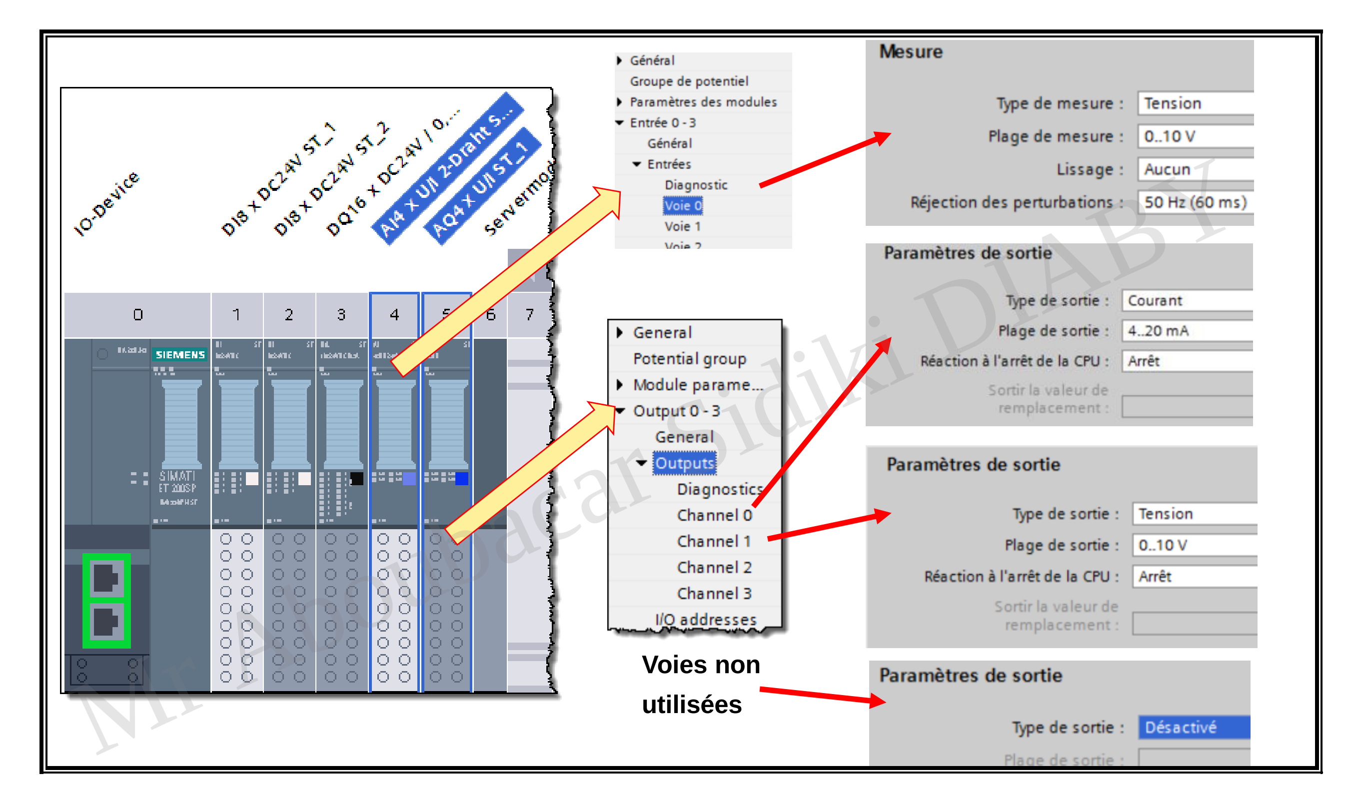
1. Sélectionnez l'éditeur « Appareils&Réseaux » la « Vues des appareils » de l'ET200SP.
2. Ouvrez le « Catalogue du matériel » au niveau des Task-Cards.
3. Configurez la station ET200SP suivant votre matériel de formation.

   Veuillez tenir compte du fait que le module AI à l'emplacement 4 nécessite une activation d'un nouveau groupe de potentiel:

4. Ouvrez la « Vue d'ensemble des appareils » (Vue en haut de page) et reportez les adresses E/S indiquées.
5. Enregistrez le projet.

14.5.6. Exercice 6 : paramétrer les voies des modules analogiques

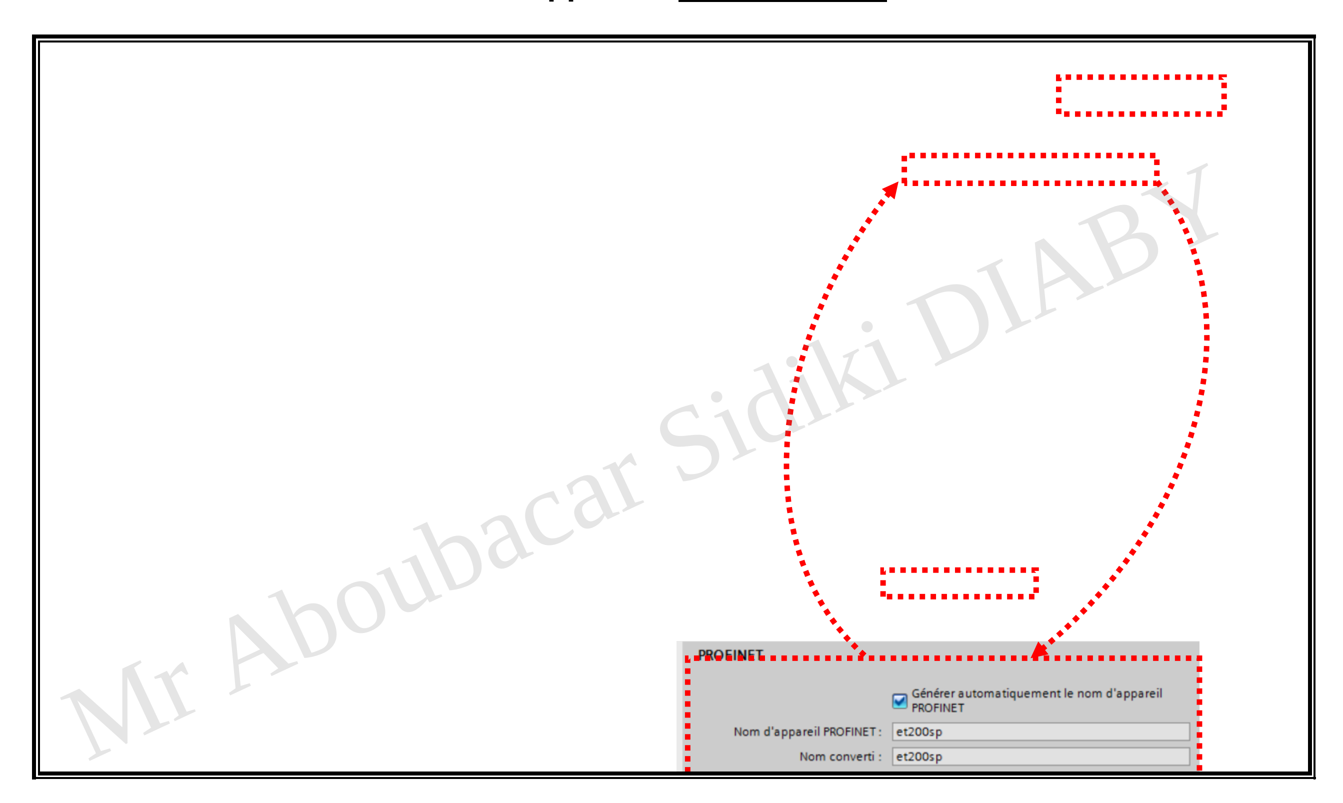


Enoncé

Paramétrez les voies analogiques comme il convient :

- Entrées périphérie décentralisée (ET200SP)
  - Voie 0 → tension 0..10 V
- Sorties périphérie décentralisée (ET200SP)
  - Voie 0 → courant 4..20 mA
  - Voie 1 → tension 0..10 V
  - Voie 2 → tension 0..10 V
- Désactivez toutes les entrées et sorties analogiques non utilisées
- En complément, activer le diagnostic de débordement haut pour les voies d'entrées analogiques de 0 à 3 car c'est un paramètre général au module

14.5.7. Exercice 7 : paramétrer l'adresse IP / le nom PROFINET de l'ET200SP et attribuer un nom aux appareils HORS_LIGNE



Enoncé

La station ET200SP doit fonctionner plus tard avec l'adresse IP, le masque de sous-réseau et le nom d'appareil PROFINET indiqués.

Le nom d'appareil PROFINET paramétré doit également être affecté en ligne à l'ET200SP afin que le contrôleur IO (l'interface PN de la CPU) puisse affecter l'adresse IP configurée lors du démarrage du système de l'ET200SP.

Marche à suivre

1. Sélectionnez la configuration des appareils de l'ET200SP

Si vous vous trouvez encore dans l'éditeur « Appareils & Réseaux », sélectionnez l'ET200SP et ouvrez l'onglet « Vue des appareils ».

Sinon, ouvrez l'éditeur « Configuration des appareils » du module périphérique décentralisé par un double-clic dans le navigateur du projet.

2. Paramétrez le nom d'appareil

Sélectionnez le module IM à l'emplacement 0 et ouvrez l'onglet « Propriétés » dans la fenêtre d'inspection. Sélectionnez ensuite l'onglet « Général » et entrez le nom d'appareil PROFINET sous « Nom ».
→ ET200SP

3. Paramétrez le nom d'appareil

Sélectionnez la rubrique « Adresses Ethernet » dans la fenêtre d'inspection et entrez sous « Protocole IP » l'adresse IP et le masque de sous-réseau.
→ Adresse IP 192.168.111.104
→ Masque de sous-réseau 255.255.255.0

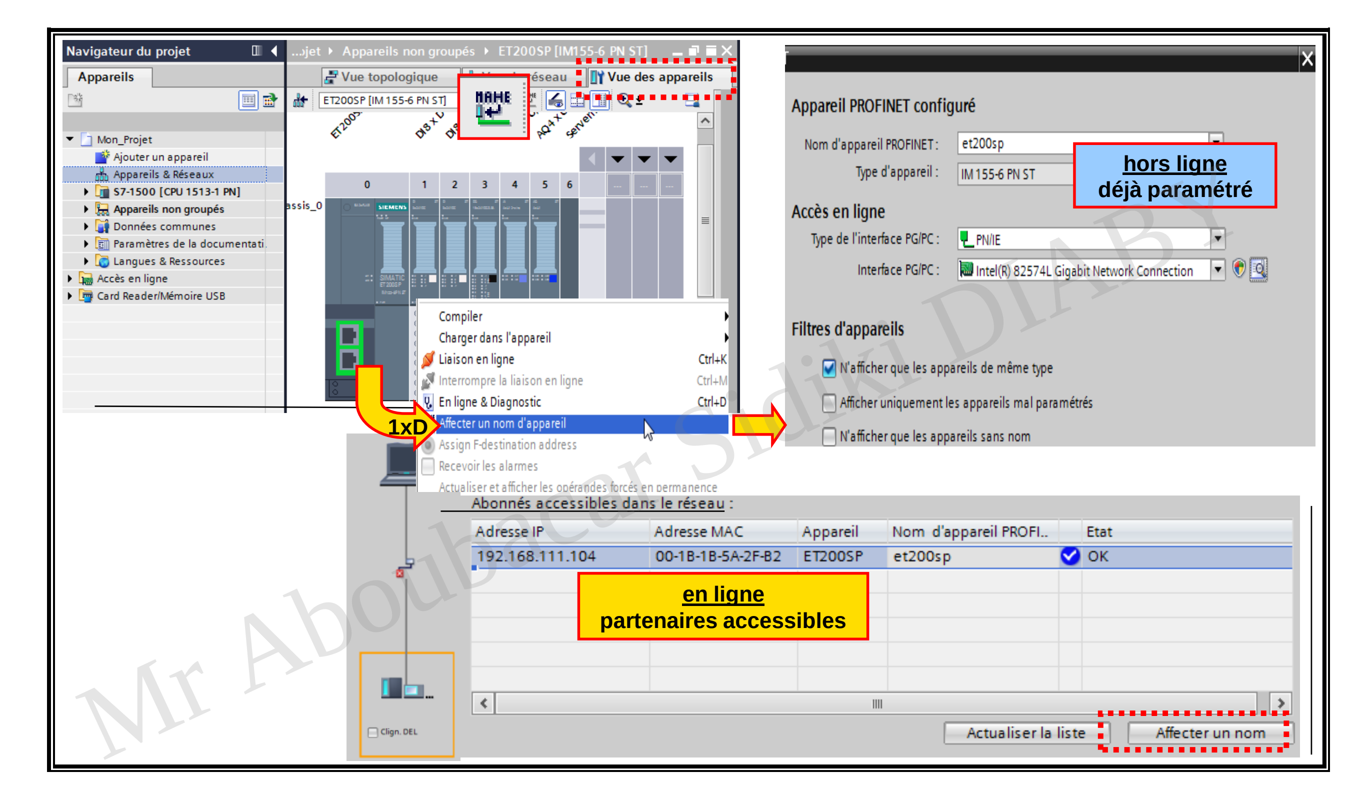
4. Complétez la vue d’ensemble des modules

Reportez le nom d’appareil et l’adresse IP paramétrés dans la vue d’ensemble de l’exercice 1 du chapitre 1.

5. Enregistrez le projet Enregistrer le projet



14.5.8. Exercice 8 : ET200SP: Attribuer un nom d'appareil EN LIGNE



Enoncé

Le nom d'appareil PROFINET précédemment paramétré doit à présent être affecté en ligne à l'ET200SP afin que le contrôleur IO ou la CPU puissent affecter l'adresse IP paramétrée hors ligne lors du démarrage du système.

Marche à suivre

1. Affectez un nom

Cliquez sur le module IM (emplacement 0) avec le bouton droit de la souris et activez l'option "Affecter un nom d'appareil" dans le menu contextuel ou utllisez le bouton. NAME

Dans la boîte de dialogue qui apparaît, vérifiez :

- le nom d'appareil PROFINET paramétré

et sélectionnez:

- sous "Accès en ligne", l'interface par laquelle vous êtes relié au PROFINET
- dans la partie inférieure de la boîte de dialogue, sous "Abonnés accessibles dans le réseau" (en ligne), l'ET200SP ou le coupleur IM155

et activez [Affecter un nom].

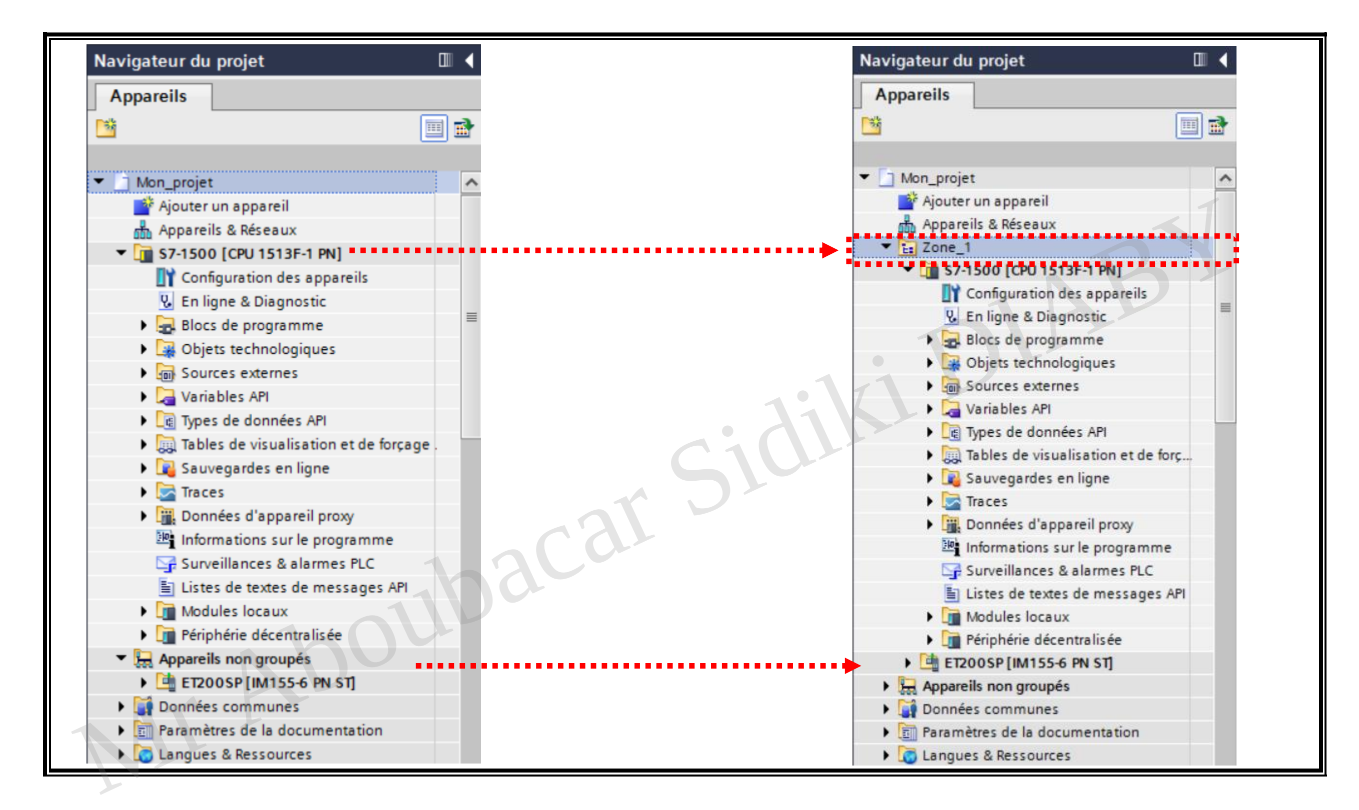
2. Fermez l'affectation du nom [Fermer]

3. Vérifiez l'affectation

Vérifiez le résultat dans la fenêtre d'inspection, onglet "Info"+"Général".

✔ Le nom d'appareil PROFINET "et200" a été affecté à l'adresse MAC "00-1B-1B-60-CB-2F" avec succès.

14.5.9. Exercice 9: Création d’un groupe et insertion des équipements



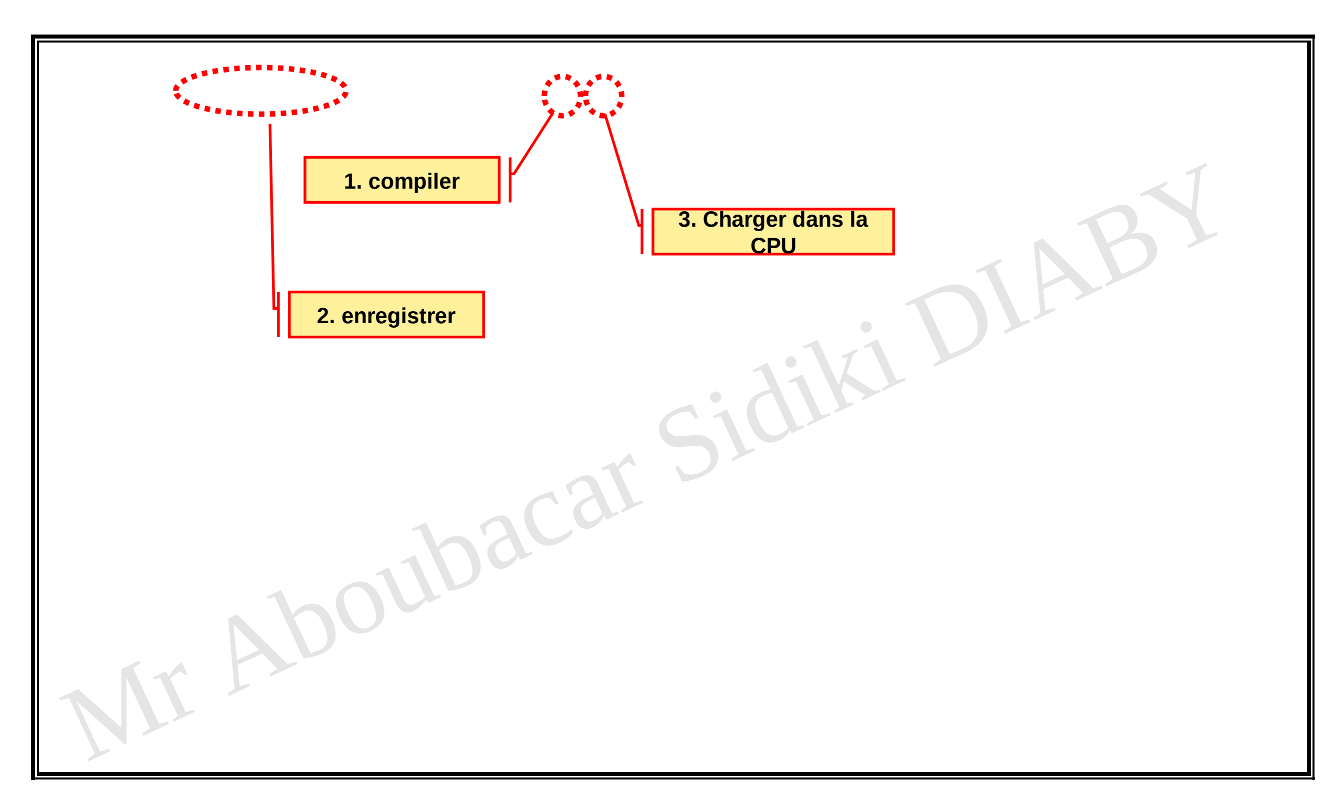
Enoncé:

Les appareils (Contrôleur et appareils d’E/S) doivent être insérés dans un groupe.

Marche à suivre:

- Sélectionnez le projet et insérez un nouveau groupe, modifiez son nom par « Zone-1 »
- Déplacez le contrôleur « S7-1500 » et l’ ET 200SP sous le répertoire « Zone-1 »
- Enregistrez votre projet.

14.5.10. Exercice 10 : compiler la configuration des appareils et la charger dans la CPU



Enoncé

Le système PROFINET I/O étant maintenant complètement configuré et paramétré, le projet doit être compilé, enregistré et chargé dans la CPU

Marche à suivre

1. Veuillez compiler la configuration matérielle en sélectionnant la station S7-1500 dans le navigateur du projet puis en cliquant sur le bouton « Compiler » (Vue ci-dessus). Veuillez vérifier dans la fenêtre d'inspection sous « Info » si la compilation s'est bien déroulée. Si des erreurs étaient apparues, veuillez les corriger.
2. Enregistrez votre projet
3. Chargez la station complète dans la CPU, en cliquant sur le bouton « Charger dans l'appareil » (Vue ci dessus). Veuillez vérifier dans la fenêtre d'inspection sous « Info » si le chargement s'est bien effectué.
4. Vérifiez les LED des modules de votre matériel de formation: seules les LED vertes doivent être allumées et non clignoter!!!
5. Enregistrez votre projet.

Résultat:

Plus que des LED vertes fixes au niveau de la CPU et de tous les modules de l'ET200SP !

14.5.11. Exercice 11 : Réassignation du programme



Le raccordement de la maquette à la périphérie décentralisée ET200SP entraîne une modification des adresses d'entrée/sortie des capteurs et actionneurs de la maquette de bande dans le programme S7. Pour rétablir le fonctionnement d'origine de l'installation, vous devez remplacer les adresses d'entrée/sortie concernées dans l'ensemble du programme S7 comme indiqué dans la figure. La méthode la plus simple consiste à utiliser la table des variables API.

Marche à suivre

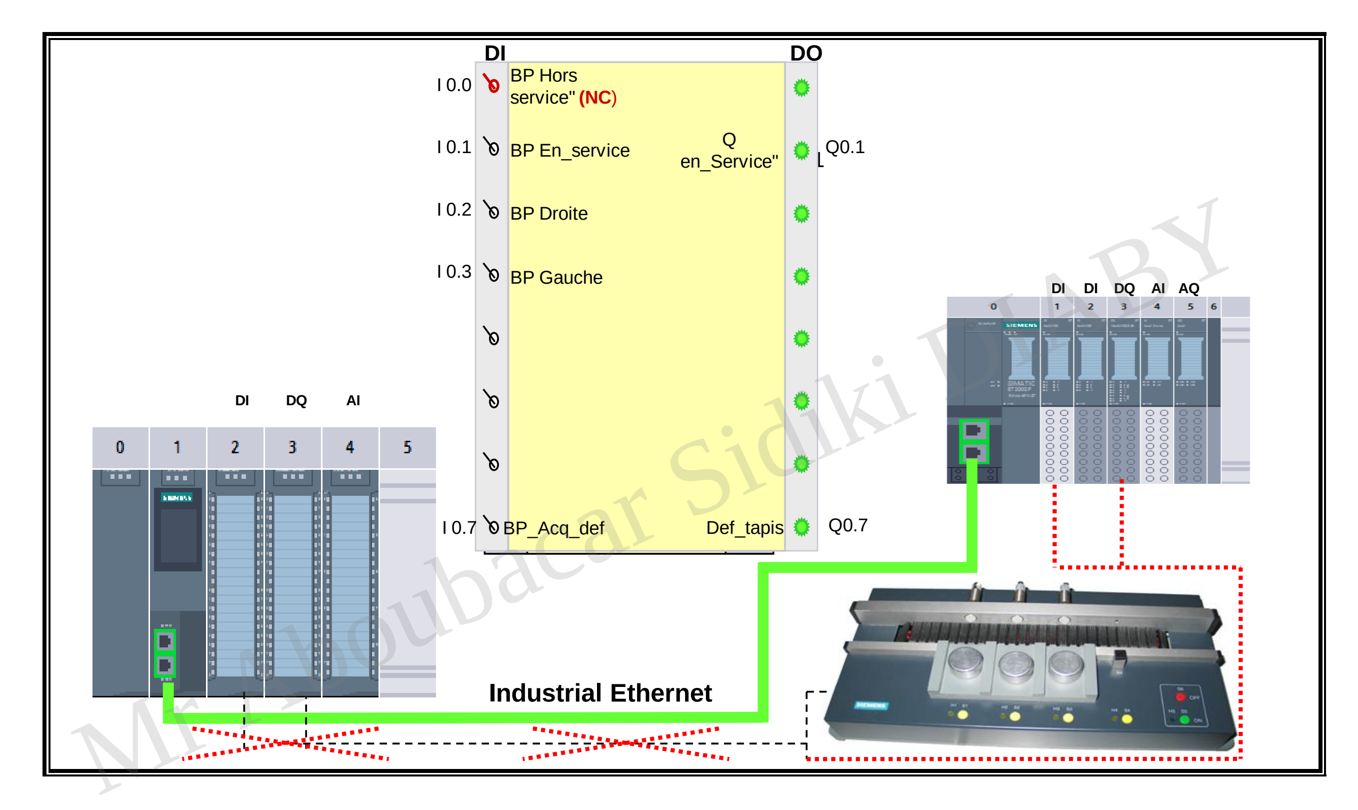
1. Ouvrez la table des variables API « Convoyeur ».
2. Modifiez les adresses d'entrée/sortie concernées comme indiqué dans la figure.

   - I3.x → I4.x

   - Q3.x → Q4.x
3. Chargez l'ensemble du programme dans la CPU.
4. Enregistrez votre projet.

Mr Aboubacar Sidiki DIABY

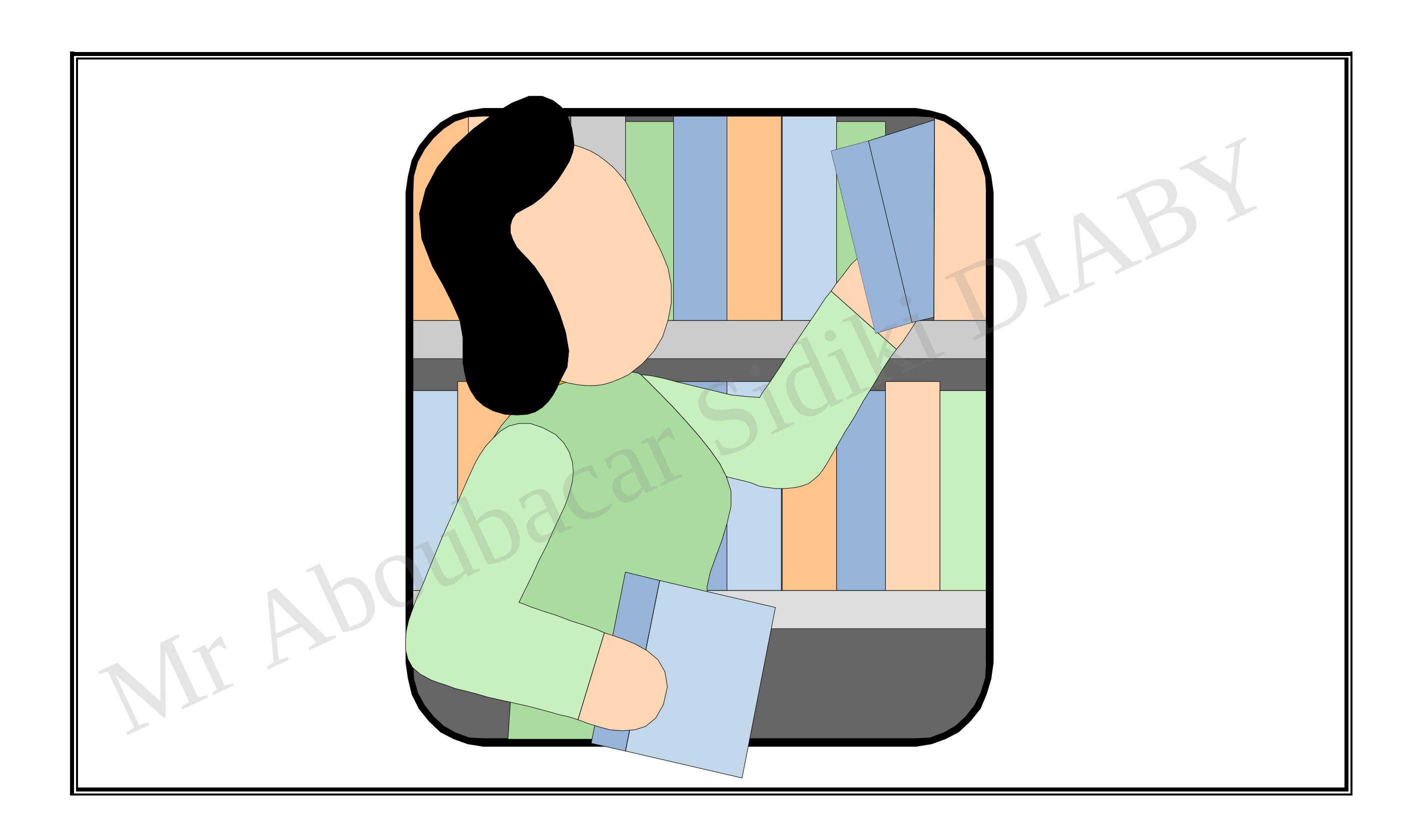
14.5.12. Exercice 12 : Test du fonctionnement avec la périphérie décentralisée



Marche à suivre

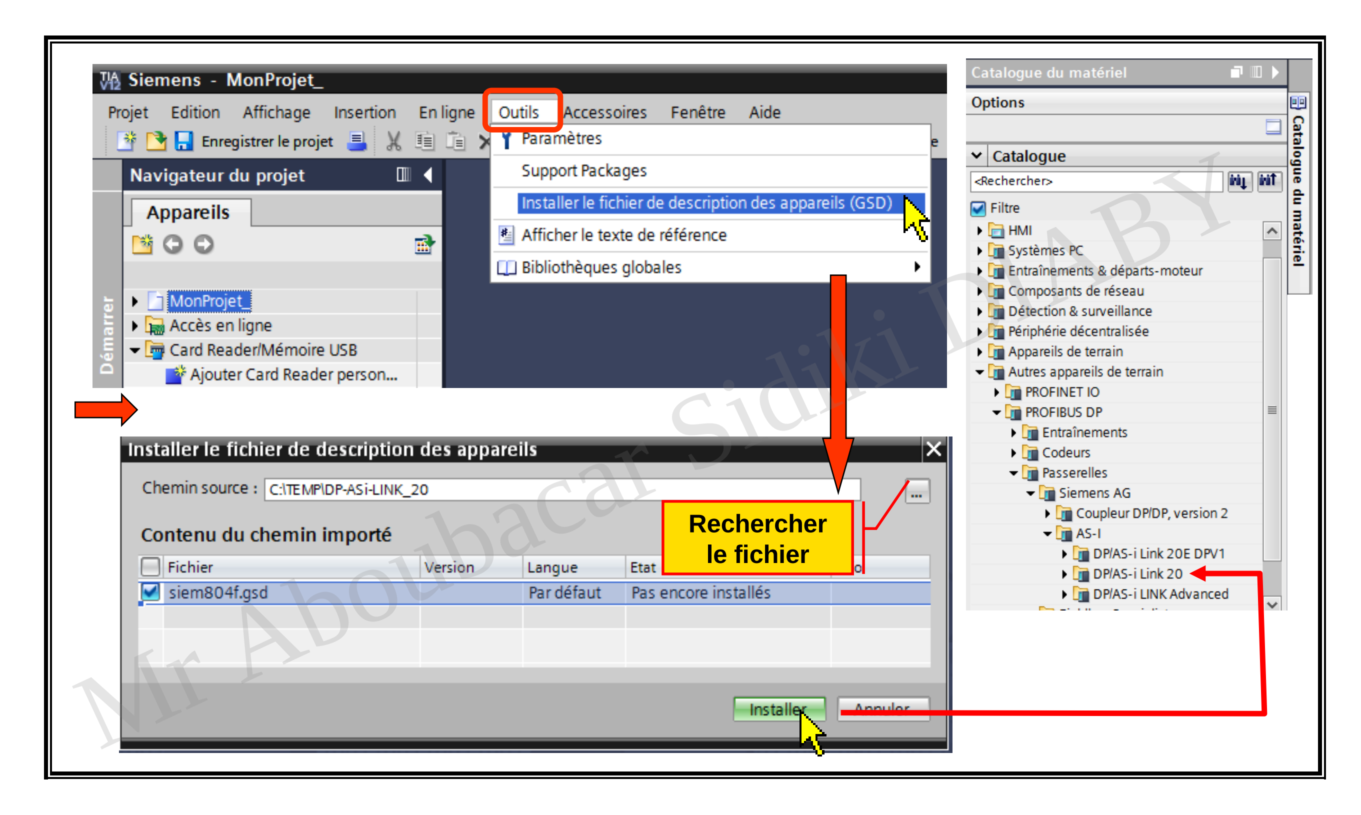
- Connectez votre maquette de simulation sur la connectique « ET200 DI/DO » à l'arrière du banc de simulation
- Mettez la maquette de simulation sous tension
- Testez le bon fonctionnement de votre programme

14.6. Informations complémentaires





14.7. Installation ultérieure de composants périphériques décentralisés



GSD – General Station Description

Pour pouvoir configurer les modules du Catalogue du matériel, STEP 7 a besoin d'un fichier GSD ou d'un fichier de type pour chaque esclave périphérique décentralisé.

Le fichier GSD décrit les principales propriétés d'un module périphérique décentralisé en conformité avec la norme PROFIBUS/PROFINET.

Il est généralement fourni avec les différents modules périphériques décentralisés ou mis à la disposition de l'utilisateur sur Internet pour téléchargement.

De nombreux modules SIEMENS sont déjà intégrés dans le Catalogue du matériel.

Modules ajoutés au Catalogue du matériel

Une fois installés, ces modules figurent dans la rubrique « Autres appareils de terrain », où ils sont disponibles pour la configuration.

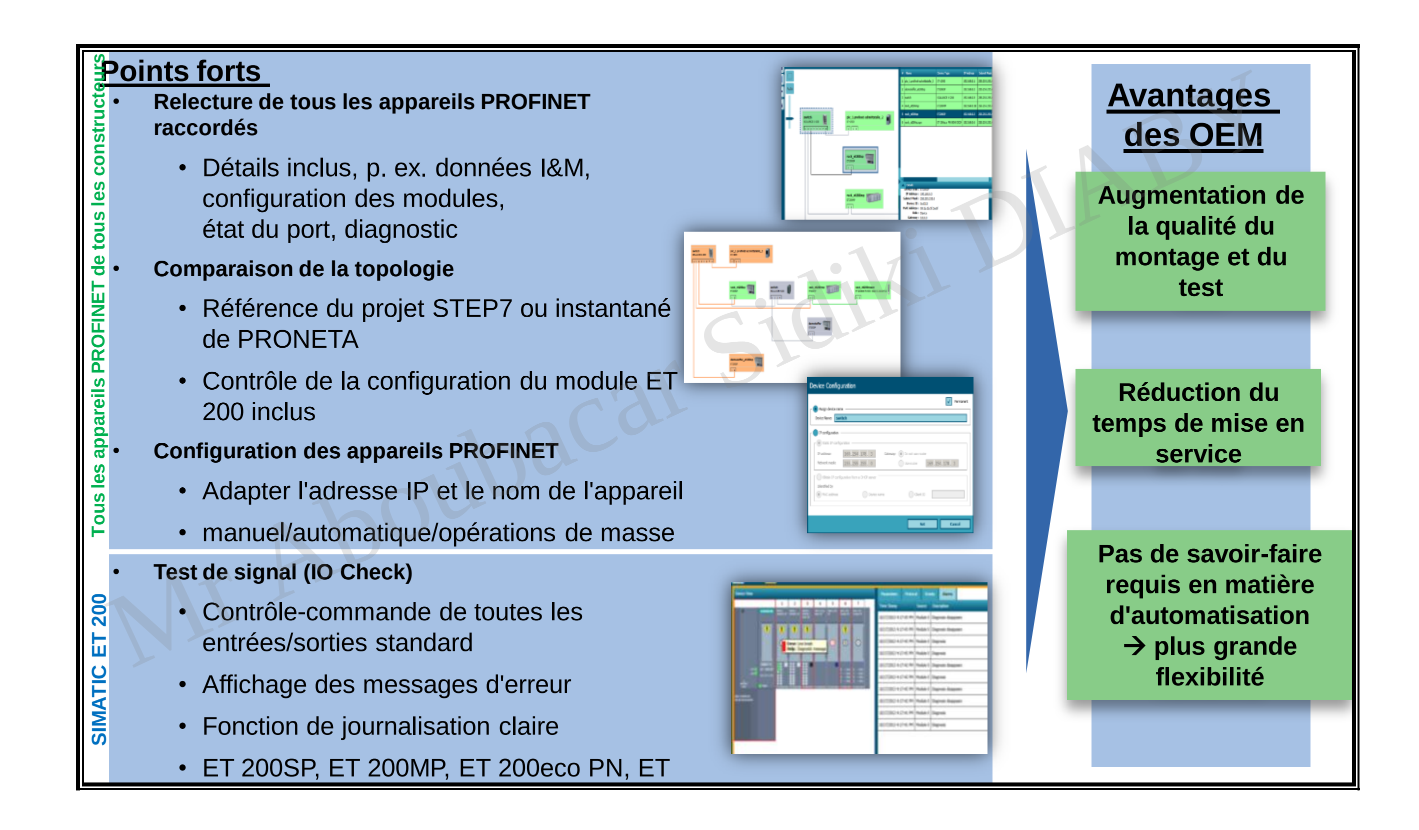
14.8. Annexe : PRONETA

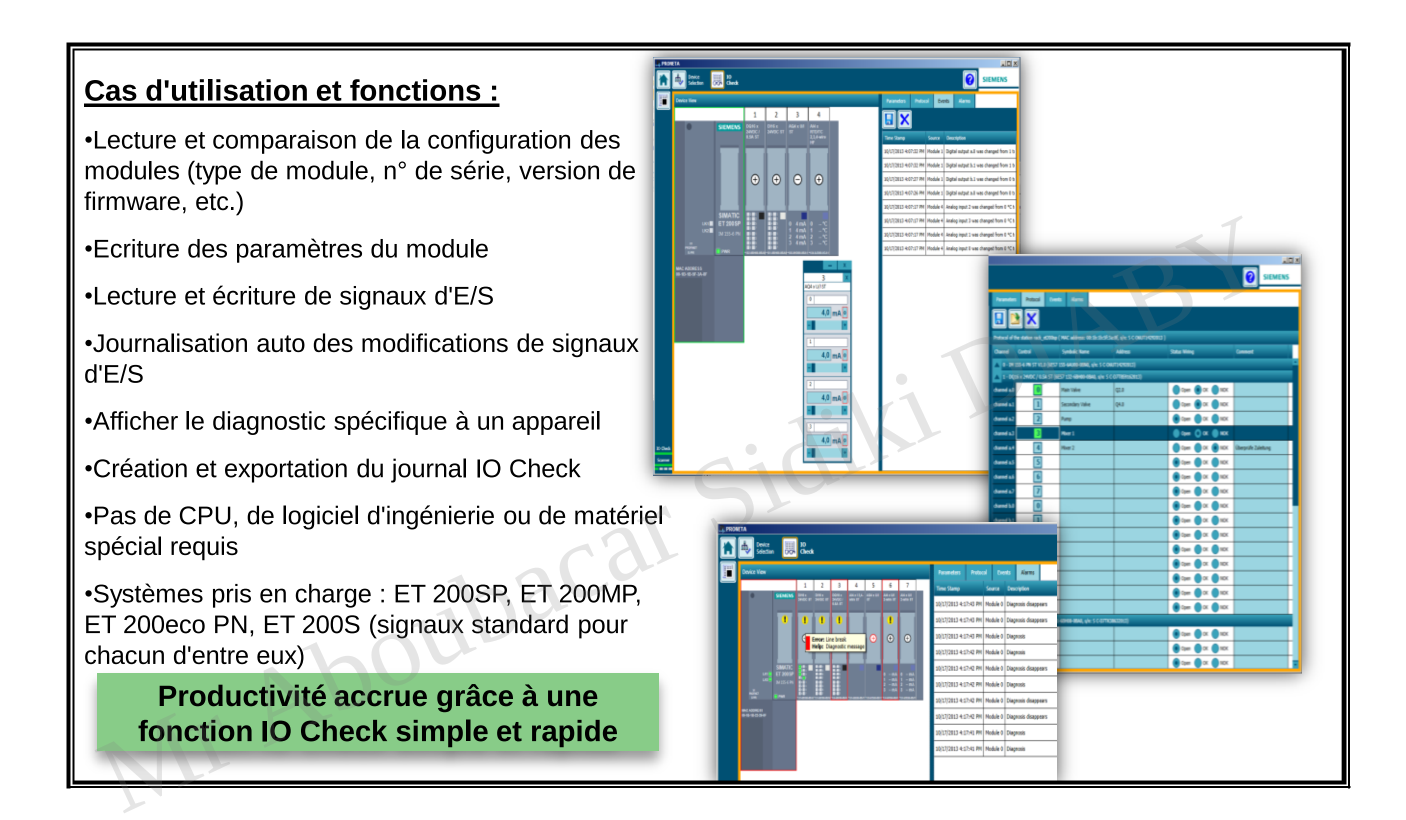


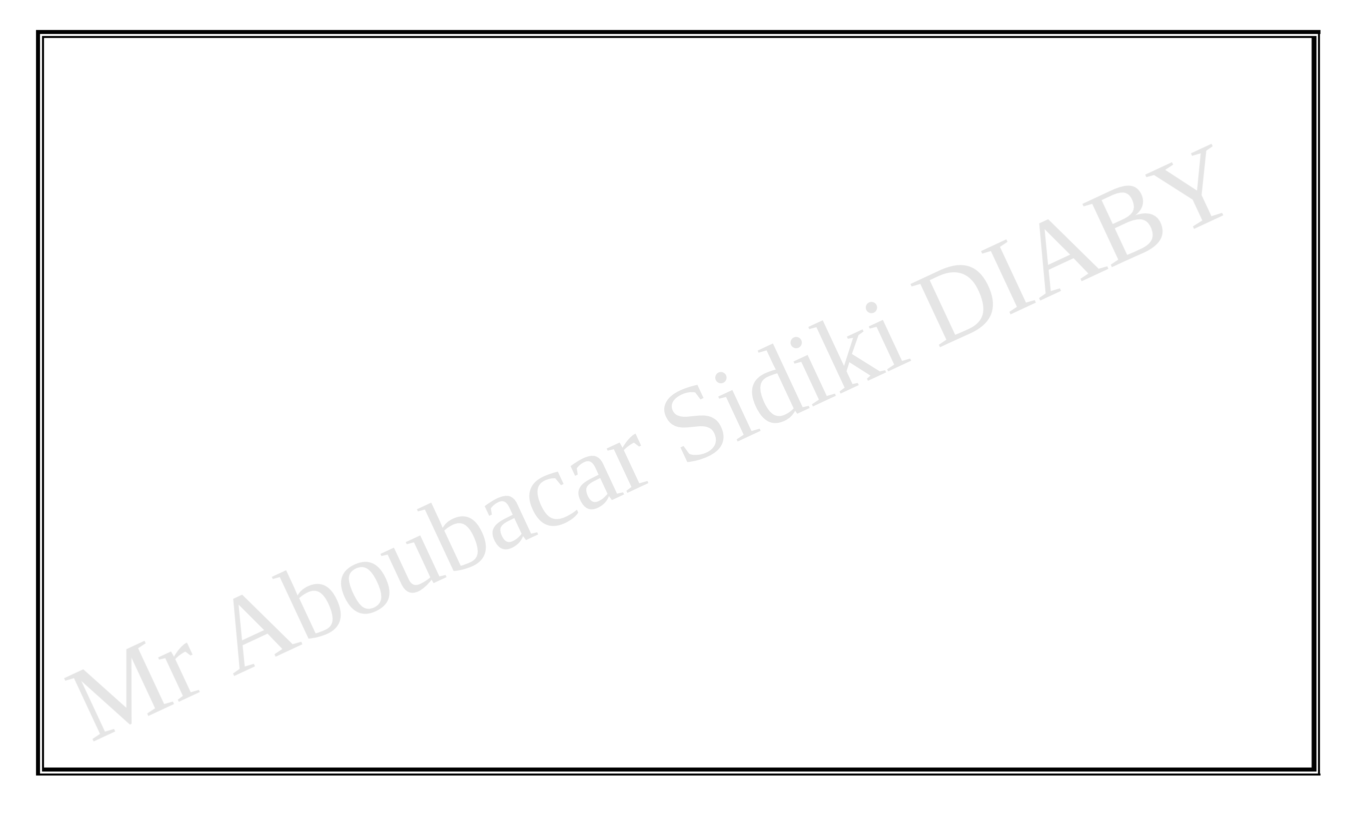
Mise en service efficace d'appareils PROFINET avec SIEMENS PRONETA

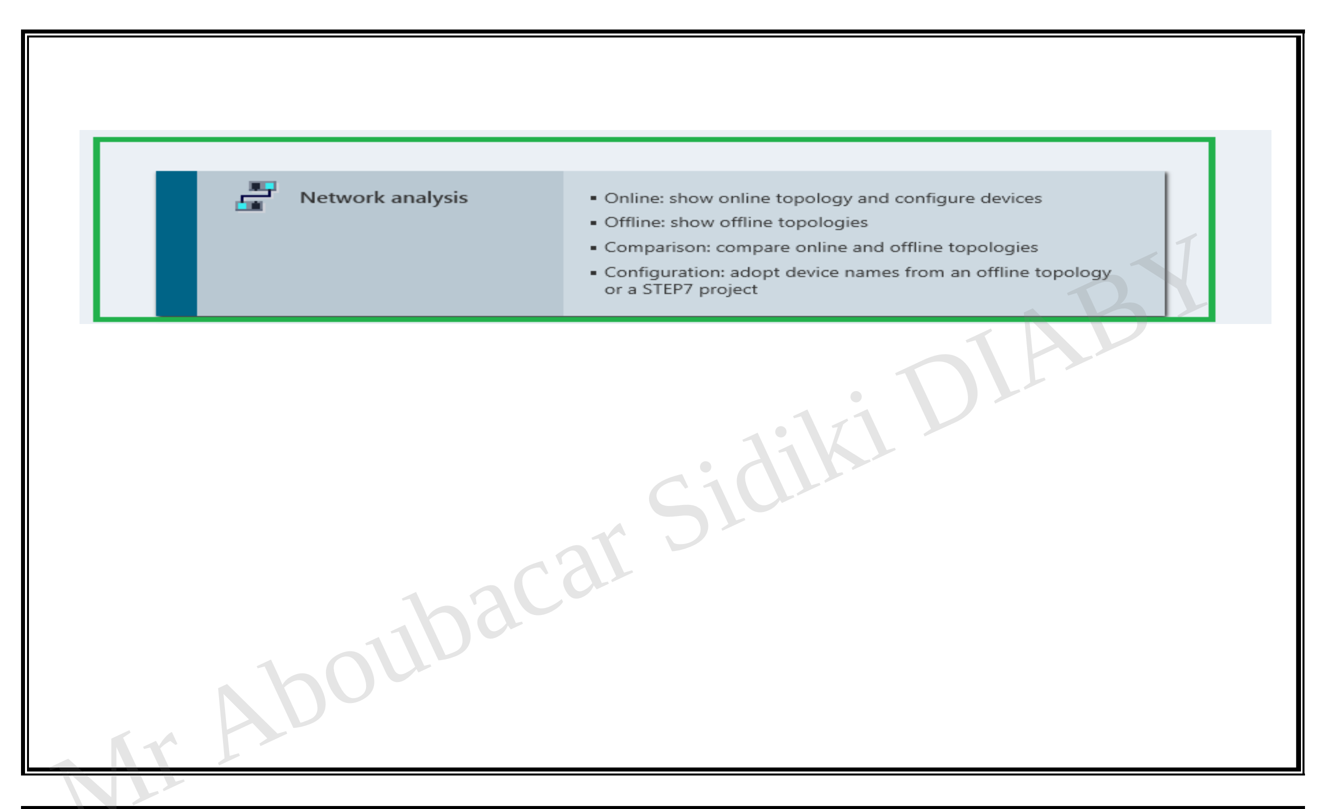
Logiciel gratuit téléchargeable sur le customer support Siemens

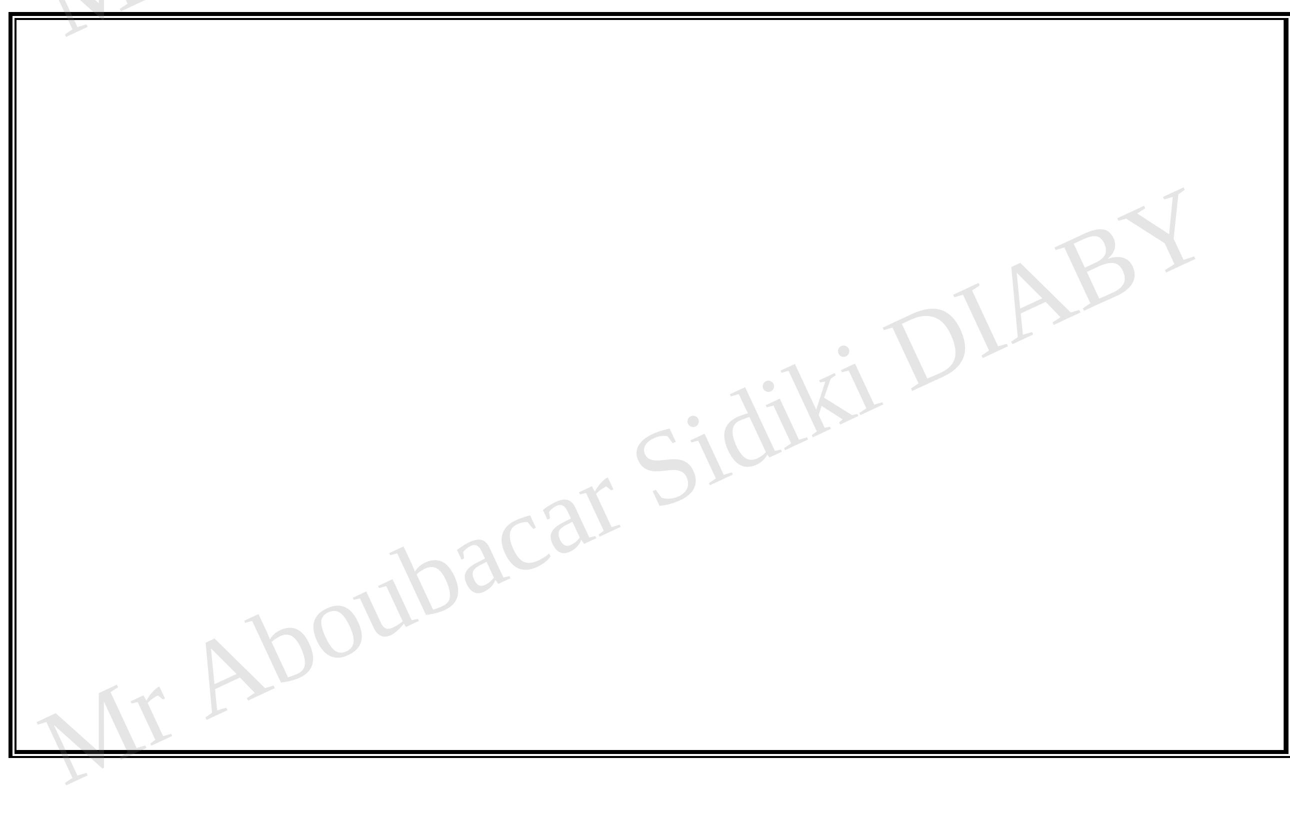


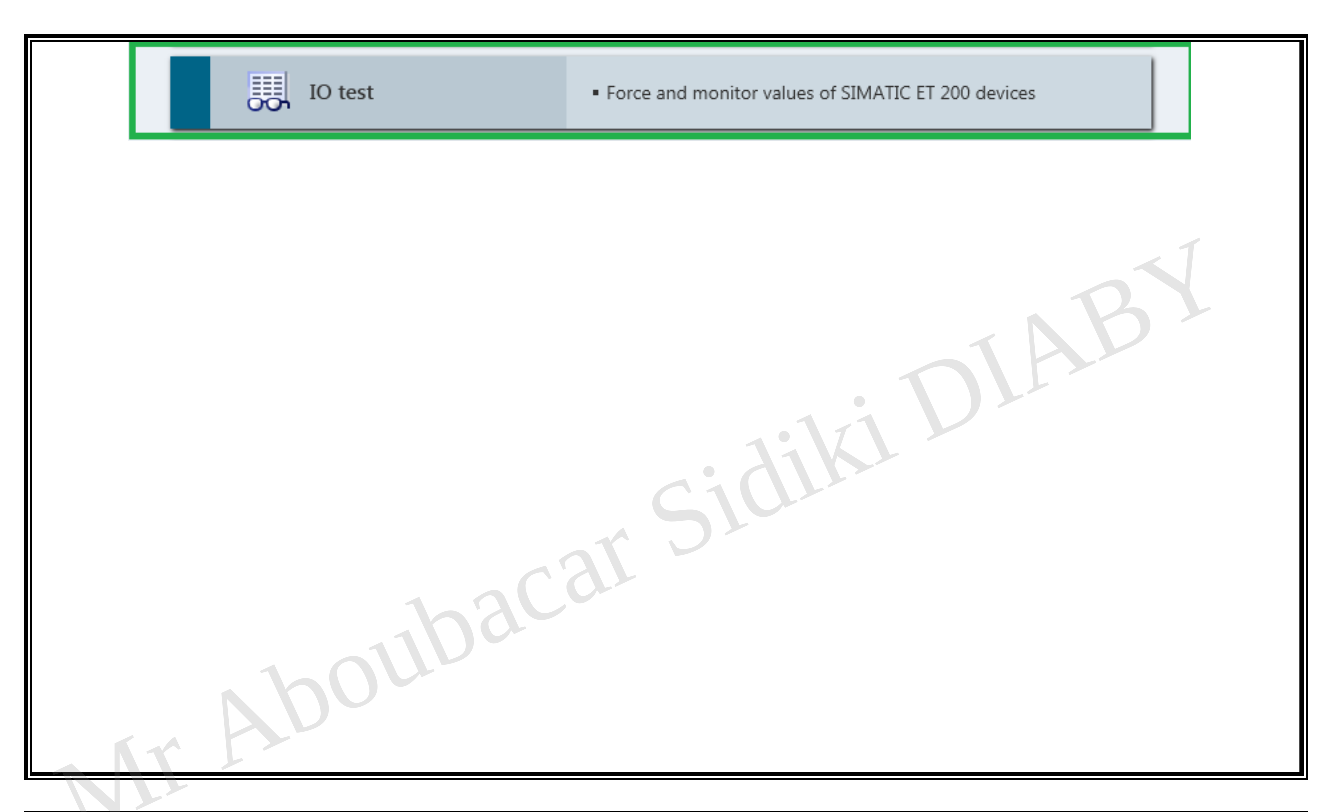


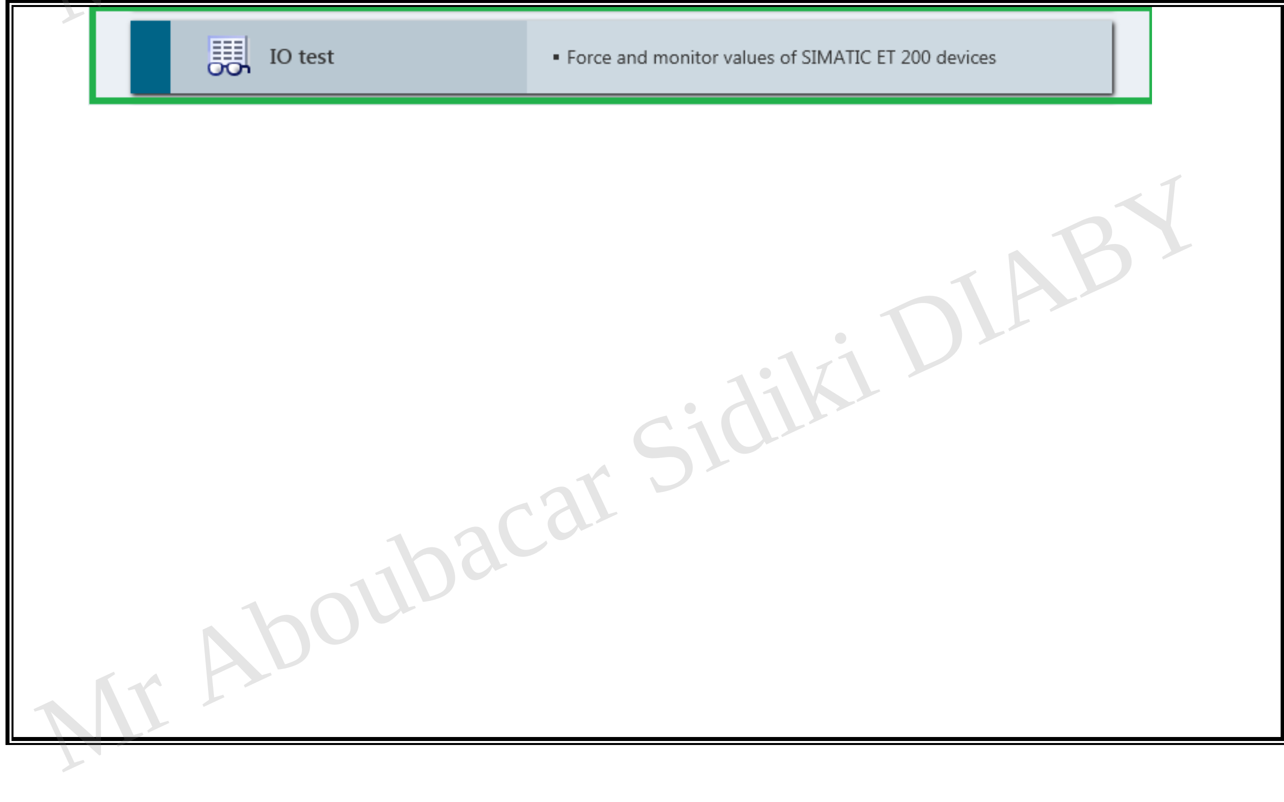

Cas d'utilisation et fonctions :
•Lecture et comparaison de la configuration des modules (type de module, n° de série, version de firmware, etc.)
•Ecriture des paramètres du module
•Lecture et écriture de signaux d'E/S
•Journalisation auto des modifications de signaux d'E/S
•Afficher le diagnostic spécifique à un appareil
•Création et exportation du journal IO Check
•Pas de CPU, de logiciel d'ingénierie ou de matériel spécial requis
•Systèmes pris en charge : ET 200SP, ET 200MP, ET 200eco PN, ET 200S (signaux standard pour chacun d'entre eux)
Productivité accrue grâce à une fonction IO Check simple et rapide
SIEMENS

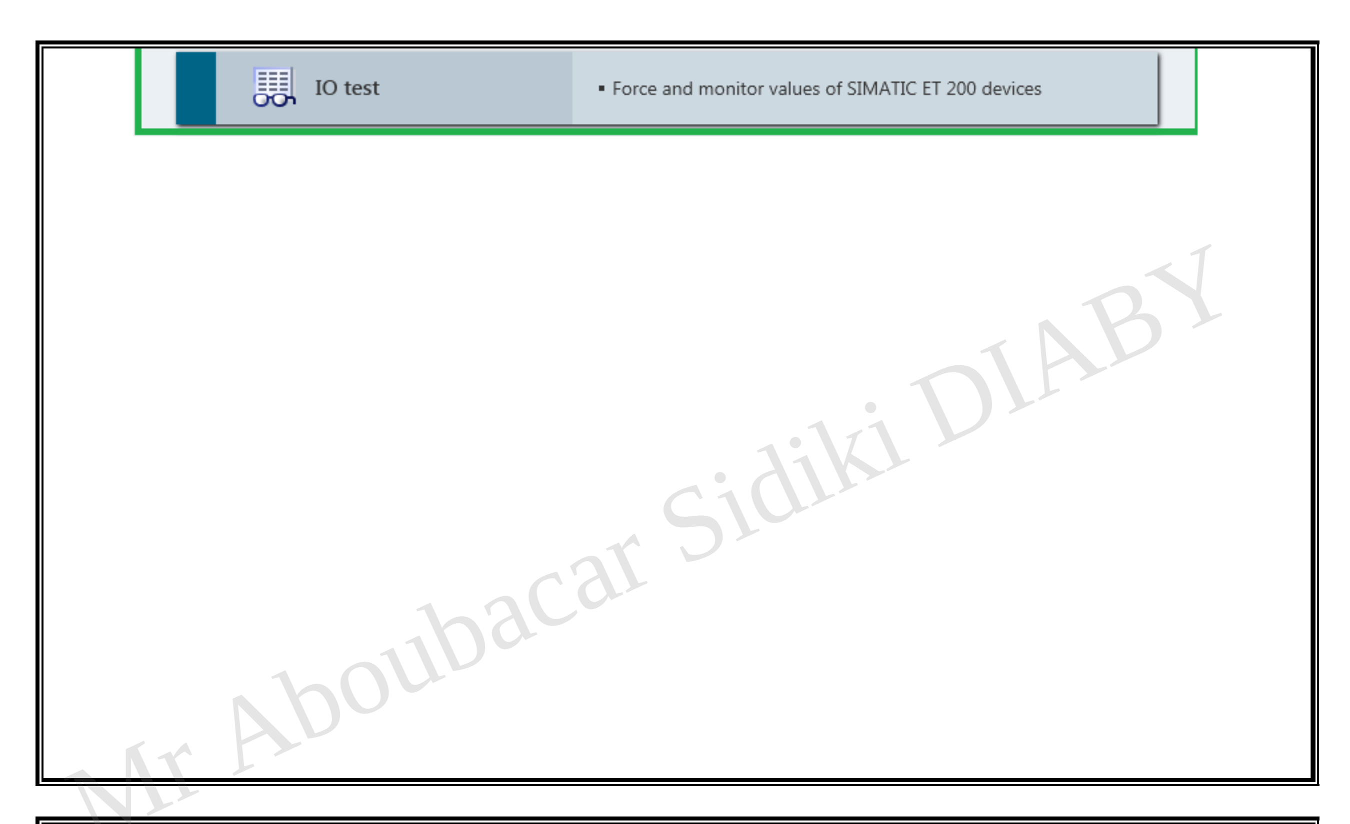


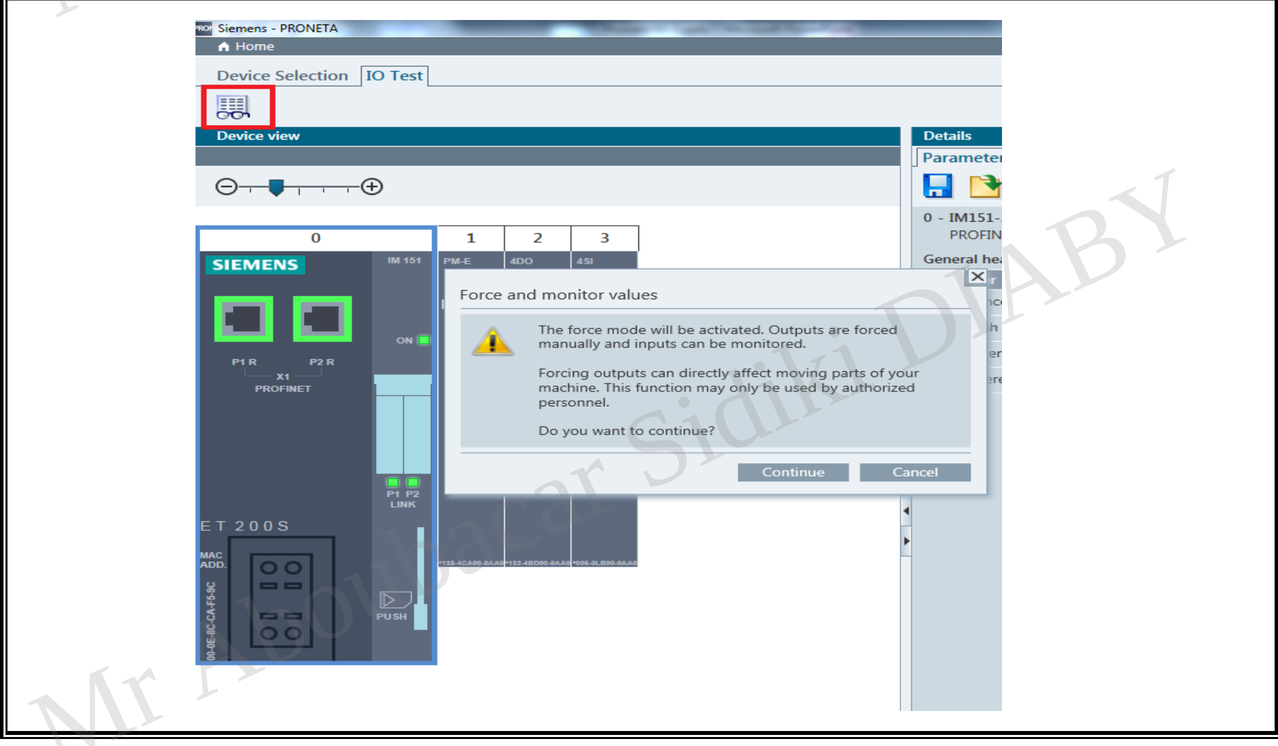

Network analysis
Online: show online topology and configure devices
Offline: show offline topologies
Comparison: compare online and offline topologies
Configuration: adopt device names from an offline topology or a STEP7 project

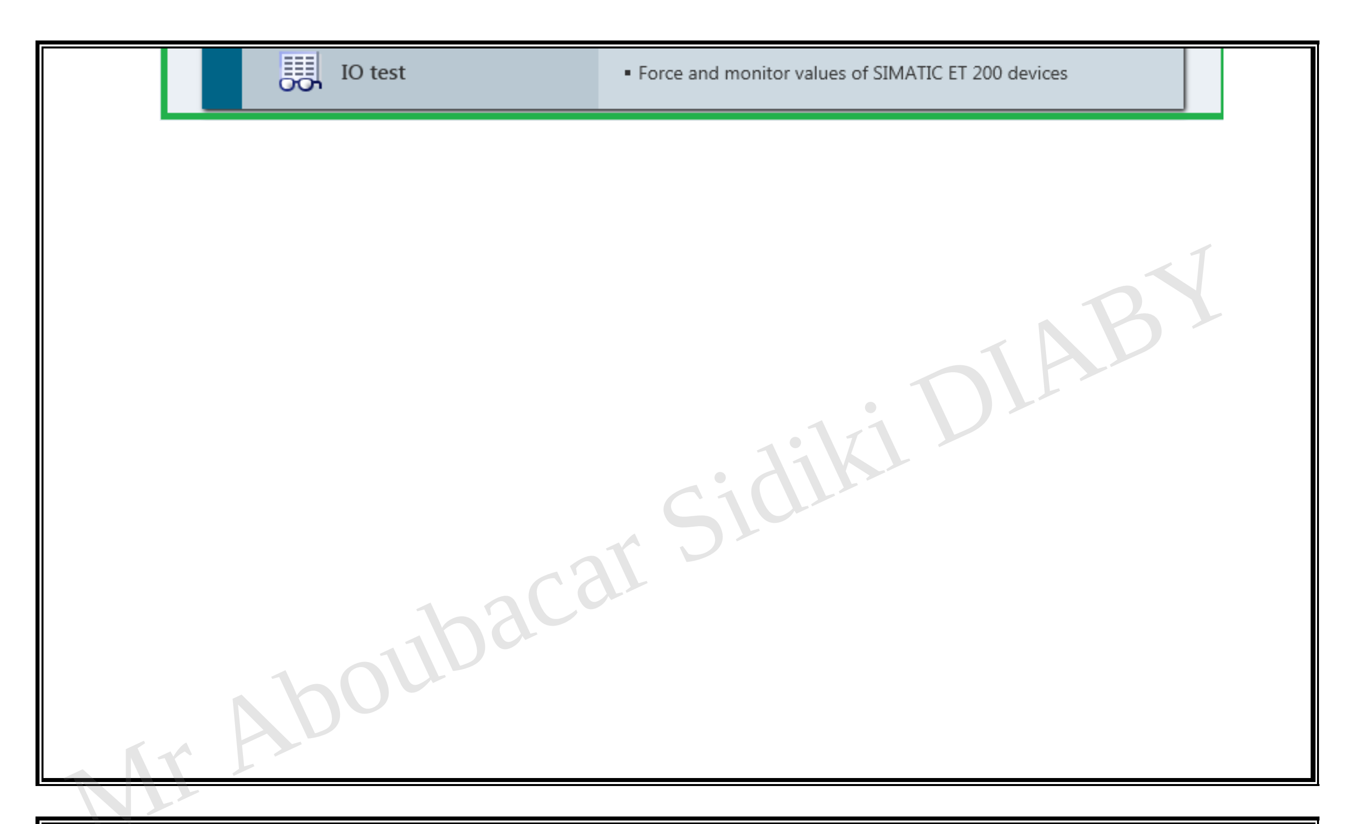


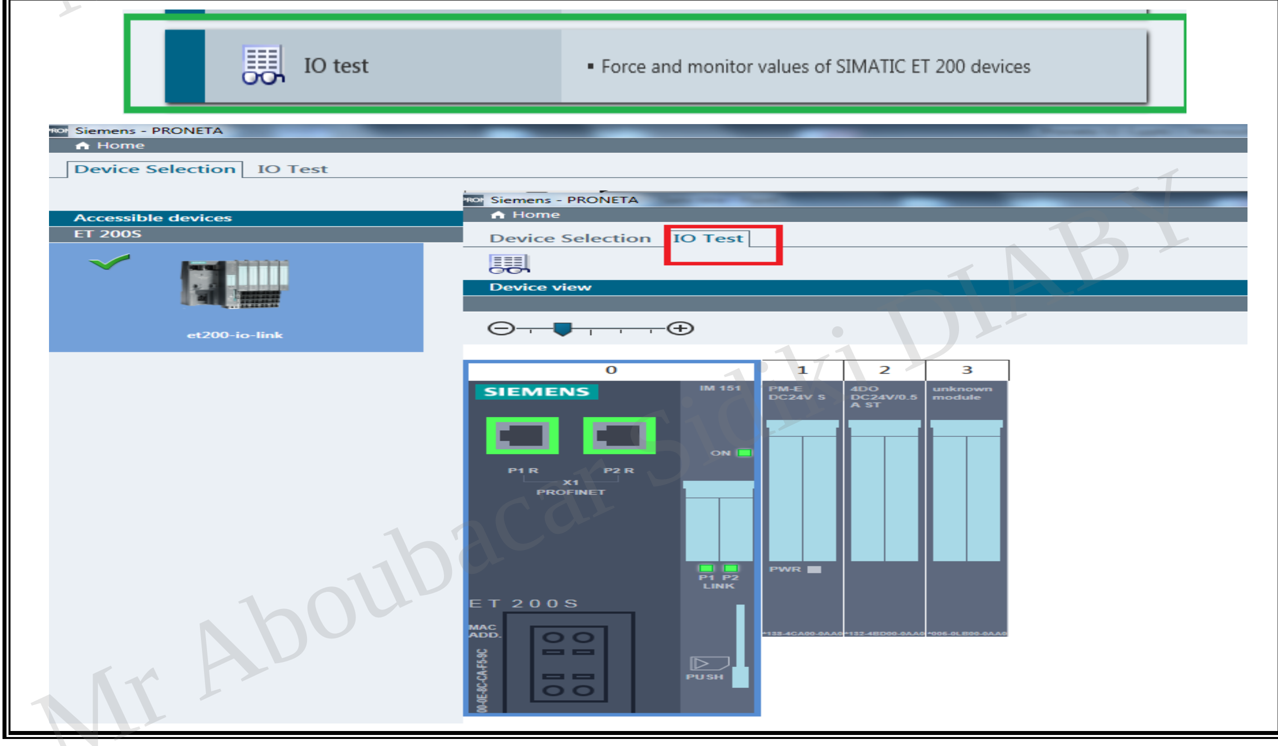

IO test
▪ Force and monitor values of SIMATIC ET 200 devices


IO test
▪ Force and monitor values of SIMATIC ET 200 devices


IO test
▪ Force and monitor values of SIMATIC ET 200 devices
Mr Aboubacar Sidiki DIABY


Siemens - PRONETA
Home
Device Selection
IO Test
Device view
Details
Parameter
0 - IM151-
PROFIN
General he
0
1
2
3
SIEMENS
IM 151
PM-E
4DO
4SI
P1 R
P2 R
X1
PROFINET
ON
P1 P2
LINK
ET 200S
MAC
ADD.
PUSH
Force and monitor values
The force mode will be activated. Outputs are forced manually and inputs can be monitored.
Forcing outputs can directly affect moving parts of your machine. This function may only be used by authorized personnel.
Do you want to continue?
Continue
Cancel
Mr Aboubacar Sidiki DIABY


IO test
Force and monitor values of SIMATIC ET 200 devices


IO test
Force and monitor values of SIMATIC ET 200 devices
Siemens - PRONETA
Home
Device Selection
IO Test
Accessible devices
ET 200S
et200-io-link
Device view
SIEMENS
P1 R
P2 R
PROFINET
ET 200S
unknown module

Sommaire

15

15. Recherche d’erreurs

A l’issue de ce chapitre vous allez

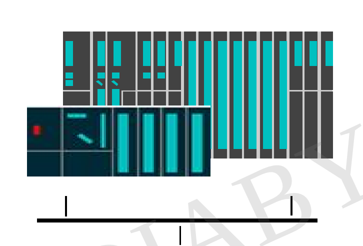
...	**connaître et savoir utiliser les fonctions de recherche d’erreurs**
…	**savoir contrôler le câblages des modules d’E/S TOR**
…	**savoir utiliser les fonctions de recherche des erreurs provoquant l’ARRET et s’en servir pour la correction des erreurs**
…	**savoir utiliser les fonctions de recherche des erreurs fonctionnelles et s’en servir pour la correction des erreurs**
…	**savoir utiliser la fonction d’analyse TRACE**

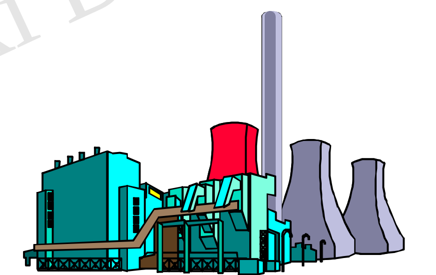


15.1. Catégories d'erreurs

Défauts détectés par le système

- Acquisition, évaluation et signalisation de défauts propres à l'API
- Défaillance d'un module
- Court-circuit sur les câbles de liaison
- Dépassement du temps de cycle
- Erreur de programmation (accès à un bloc inexistant)





Défauts fonctionnels

- Le fonction requise n'est pas du tout assurée ou est exécutée incorrectement
- Défauts processus (capteur/actionneur, câbles défectueux)
- Erreur logique dans le programme (non détectée à la programmation et à la mise en route)

Fonctions de surveillance

Le diagnostic est important pour l'exploitation d'une installation ou d'une machine. On fait généralement un diagnostic lorsqu'un défaut provoque l'arrêt ou un dysfonctionnement d'une installation ou d'une machine. Etant donné les coûts occasionnés par les temps d'arrêt ou les dysfonctionnements, il faut trouver rapidement l'origine du défaut et l'éliminer.

Catégories d'erreurs

Les erreurs peuvent se classer en deux catégories, selon qu'elles soient ou non détectées par l'API:

· Les défauts détectés par le système d'exploitation de l'API et provoquant normalement l'arrêt de la CPU.

Les défauts fonctionnels c. à d. que la CPU poursuit le traitement du programme, mais la fonction retenue ne s'exécute pas du tout ou s'exécute de manière erronée. La recherche de ce type d'erreurs est généralement beaucoup plus difficile, car dans un premier temps, on ignore ce qui a provoqué le dysfonctionnement.

Causes possibles:

· Une erreur de logique dans la programmation (erreur logicielle), qui n'a pas été détectée à l'élaboration et à la mise en service du programme utilisateur et qui ne se produit que dans de très rares états de fonctionnement de l'installation.

· Un défaut dans le processus, déclenché par un dysfonctionnement d'un des composants de la chaîne de commande, comme les câbles des capteurs et des actionneurs ou encore par un défaillance des capteurs et des actionneurs eux-mêmes.

15.2. STEP 7 – Vue d'ensemble des fonctions de test

Erreurs détectées par le système
En général: CPU en Run

- Erreur de programmation (sans OB121): **STOP**
- Erreur d'accès (sans OB122): **RUN**
- Alarmes de diagnostic (sans OB82, 83, 86): **RUN**

- **En ligne & Diagnostic**
  - Tampon de diagnostic
- **Task Card "Tester"**
  - Hiérarchie d'appel
  - Registres de la CPU
  - Pile de données locales (en préparation
- **Diagnostiquer des modules**
  - Diagnostic d'état (pour tous les modules)

Erreurs fonctionnelle
En général: CPU en RUN

- Défaut processus (par ex. rupture de fil)
- Erreur logique dans la programmation (par ex. double affectation)

- **Visualisation et forçage de variable**
  ® Tables de visualisation et de forçage permanent
- **Visualisation de blocs (Etat du bloc)**
  ® Visualisation dans l'éditeur de blocs
- **Outils**
  - Listes de références croisées
  - Tableau d'affectation des opérandes (I/Q/M/T/C)
  - Structure d'appel

- **Fonction analyse "Trace"**
- **Comparaison de programmes**
- **Définition de points d'arrêt**

En préparation pour S7-1200/1500

Fonctions de test

STEP 7 propose différentes fonctions de test adaptées à la nature de l'erreur à rechercher:

· **...si la CPU est à l'ARRET**

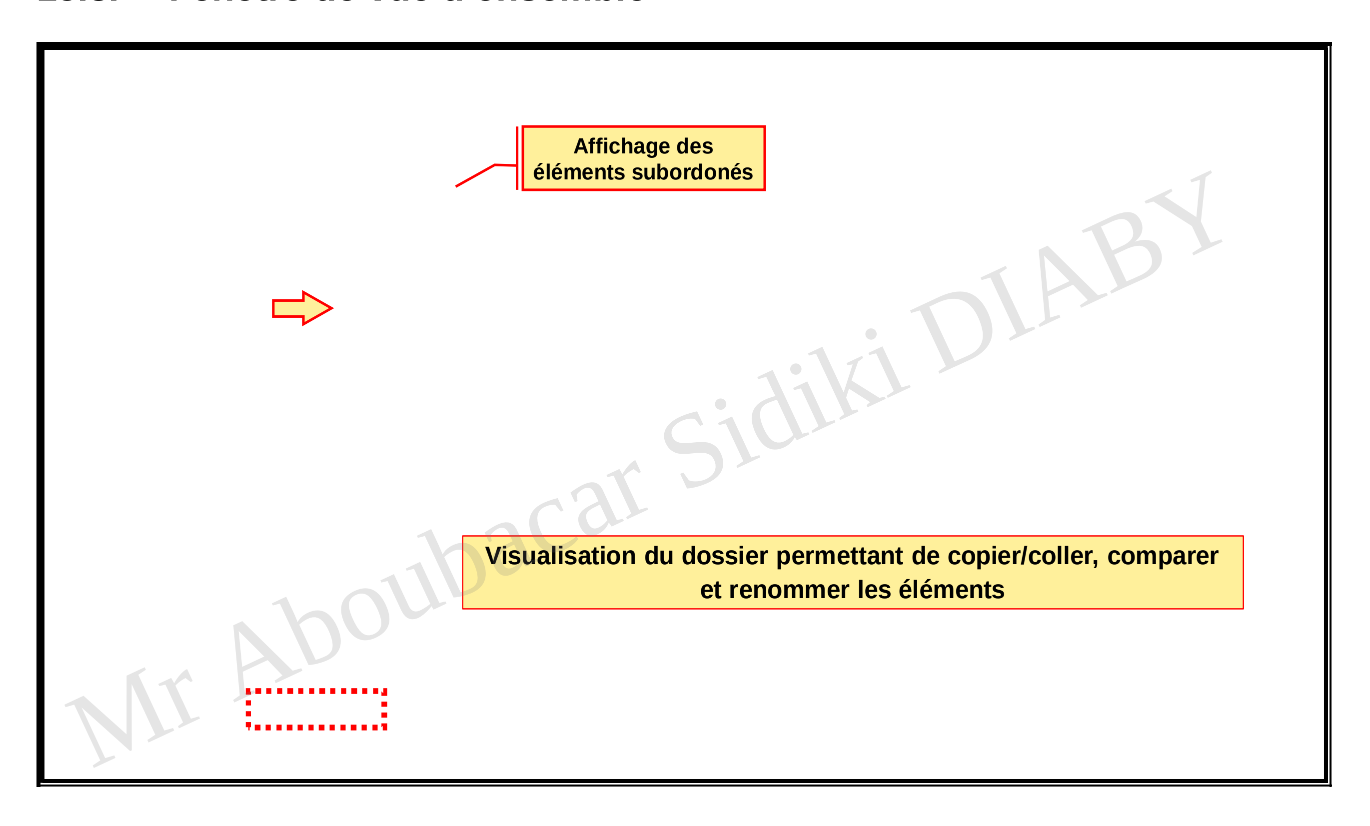
Les défauts détectés par le système sont traités à l'aide des fonctions de test: Mémoire tampon de diagnostic, Pile I, Pile B, Pile L et Diagnostic matériel qui fournissent des informations détaillées sur la cause de l'erreur et sur le point d'interruption. La programmation d'OB d'erreur (voir chapitre Blocs d'organisation) permet, par programme, d'analyser l'erreur survenue et d'éviter la mise à l'arrêt de la CPU. Etant donné que la CPU n'enregistre ou ne délivre à l'arrêt plus de mémoire image du processus et ne poursuit pas l'exécution du programme, il n'est pas judicieux d'utiliser les fonctions de test « Visualisation/forçage des variables » et « Visualisation du bloc ».

· **...si la CPU est en MARCHE**

Inversement il est généralement peu judicieux d'utiliser, sur une CPU en RUN, les fonctions de test comme Pile I, Pile B ou Pile L pour la recherche d'erreur, car l'exécution du programme n'a pas été interrompue et que le système ne fournit aucune information relative à l'erreur survenue. La fonction de test « état du module » permet en fait d'obtenir des informations générales sur l'état de la CPU ou sur des erreurs antérieures. Le diagnostic des défauts fonctionnels s'effectue de la manière suivante:

- **Défauts liés au processus** (par ex. erreur de câblage)
  Test de câblage des entrées: *Visualiser les variables*
  Test de câblage des sorties: débloquer les sorties périphériques *(uniquement avec* CPU à l'arrêt)
- **Erreurs de programmation logiques** (par ex. double affectation)
  La recherche d'erreurs de programmation logiques fait appel à l'ensemble des fonctions de test présentées excepté la fonction débloquer les sorties périphériques.
- **Forçage:** forçage des opérandes indépendamment de la logique du programme
- **Point d'arrêt:** Suivi de l'exécution du programme pas à pas

15.3. Fenêtre de vue d'ensemble



La fenêtre de vue d'ensemble est utilisée pour compléter l'arborescence du projet. La fenêtre de vue d'ensemble affiche les contenus des répertoires sélectionnés dans le navigateur du projet.

Vous pouvez également exécuter les actions suivantes au niveau de la fenêtre :

- Ouvrir des objets
- Afficher et éditer les propriétés des objets dans la fenêtre d'inspection
- Renommer des objets
- Exécuter des actions spécifiques aux objets dans le menu contextuel
- Comparer des objets
- Exécuter différents tâches liées aux objets comme, par exemple, insérer des objets de la bibliothèque en utilisant glisser et déposer, déplacer, copier, insérer et effacer des objets

Il est possible de comparer les contenus de deux répertoires ou de deux objets. Pour cela, la fenêtre peut être partagée. De plus, il est possible de déplacer des objets entre les deux fenêtres partagées en utilisant glisser et déposer.

15.3.1. Recherche globale, Rechercher / Remplacer dans l'éditeur

Zone de recherche globale au projet

Editeur poure rechercher / remplacer

Limite de la recherche

Résultat

Dans TIA Portal, vous pouvez utiliser les possibilités de recherche suivantes :

- Recherche dans l'ensemble du projet (barre des tâches ou CTRL+F lorsque le focus n'est pas dans la fenêtre (de travail) de l'éditeur
- Rechercher et remplacer dans un éditeur (task card : Tâches > Rechercher et Remplacer
- Rechercher dans le catalogue du matériel

Recherche dans le projet

Vous pouvez rechercher un texte spécifique dans l'ensemble d'un projet. Pour cela, vous disposer d'un éditeur de recherche dans lequel vous pouvez, par exemple, limiter la recherche. Les objets qui contiennent le texte recherché sont clairement présentés dans un tableau. Vous pouvez, à partir de l'éditeur de recherche, ouvrir chaque objet afin de vérifier son emplacement.

La recherche globale (recherche dans un projet) peut être démarrée par CTRL+F (quand le focus/sélection ne se trouve pas dans la zone (de travail) de l'éditeur ou via le menu contextuel au niveau du nom du projet ou encore via the menu Edition > Rechercher dans le projet.

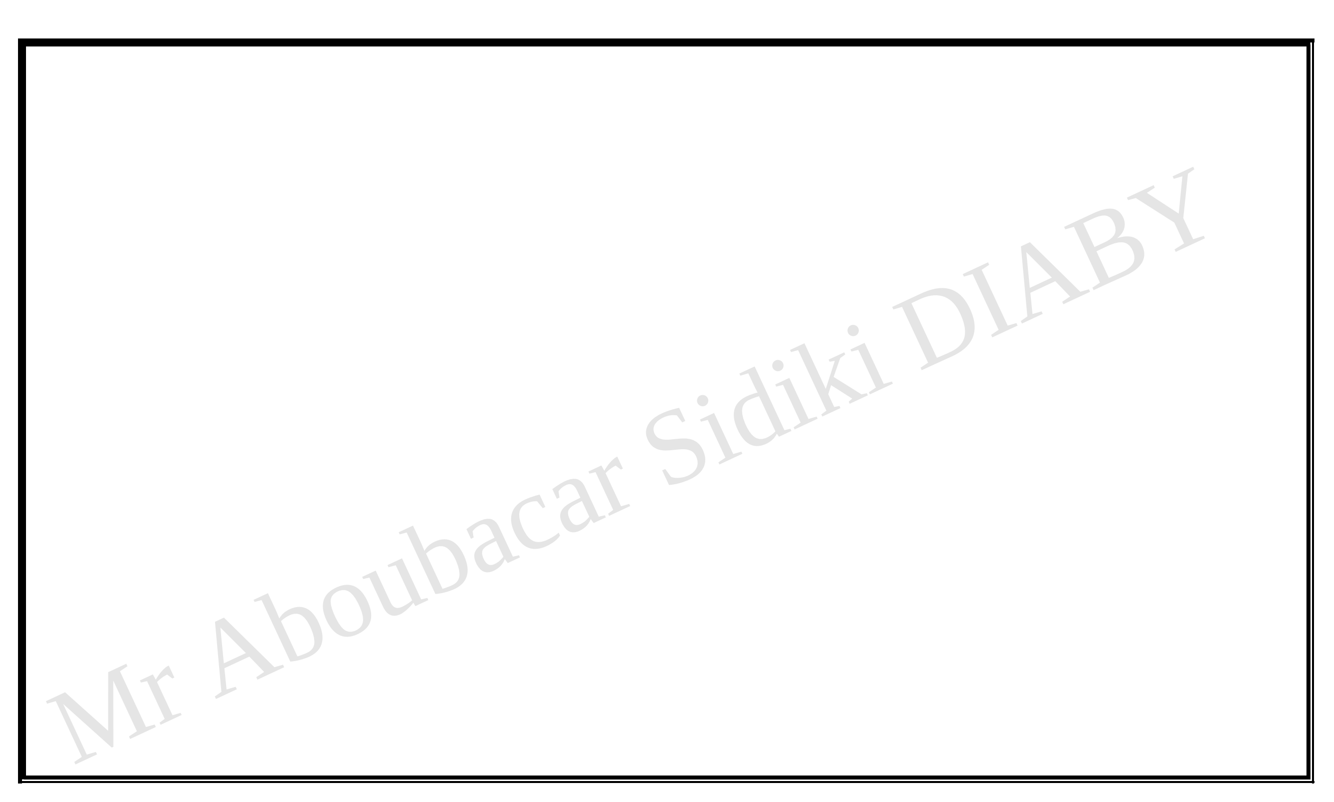
Rechercher et remplacer dans un éditeur (CTRL+F le focus est dans la fenêtre de travail de l'éditeur)

Il est possible de rechercher des textes dans les éditeurs. La fonction de recherche trouve tous les textes dans l'éditeur ouvert contenant le terme recherché. Les résultats sont sélectionnés l'un après l'autre au niveau de l'éditeur ouvert.

En complément, vous disposez des possibilités suivantes:

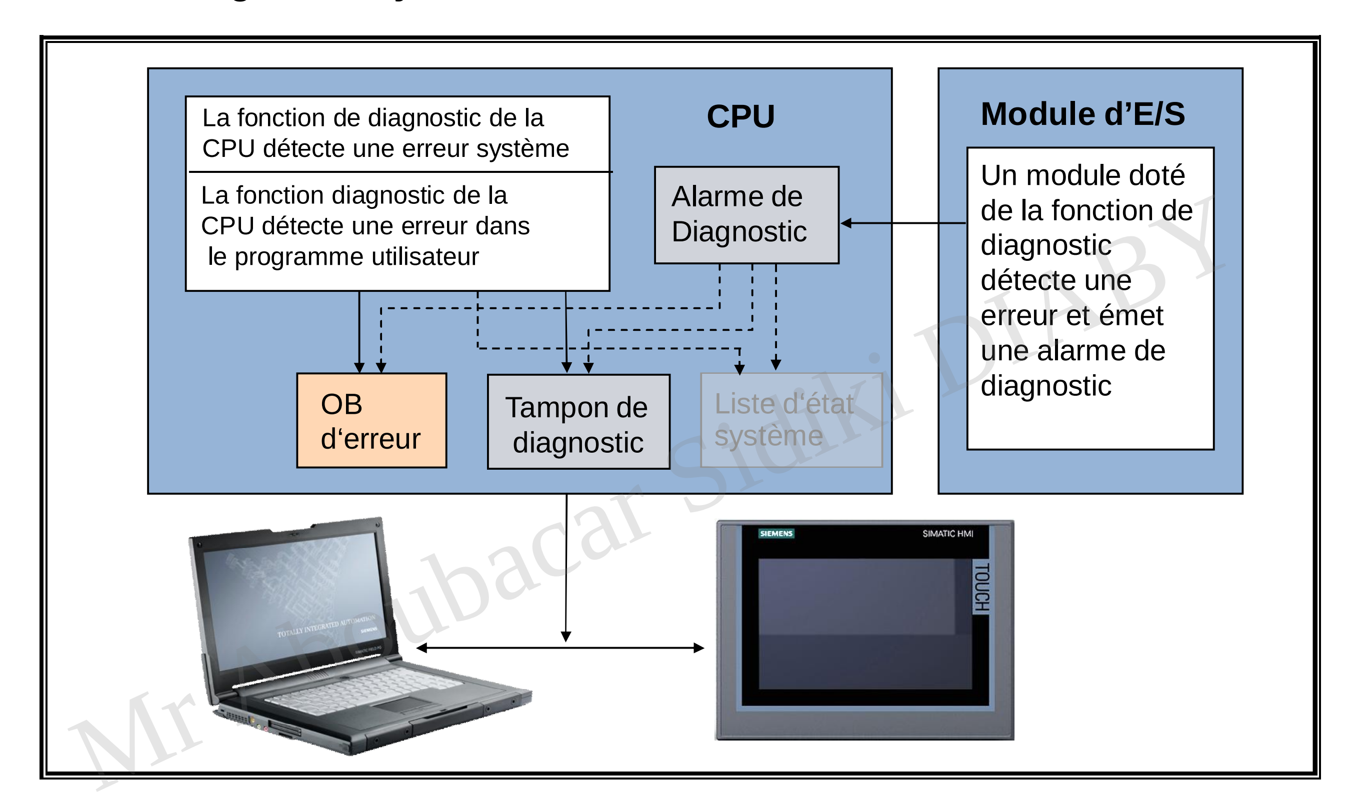
- Approfondir la recherche par des options
- Remplacer les instances des textes.

15.4. Information détaillée dans la vue d'ensemble des adresses



Dans le menu « Vue d'ensemble des adresses » des propriétés de la CPU, une vue d'ensemble exhaustive des adresses d'E/S centralisées et des périphéries peut être visualisée.

15.5. Diagnostic système – Vue d'ensemble



Diagnostic système

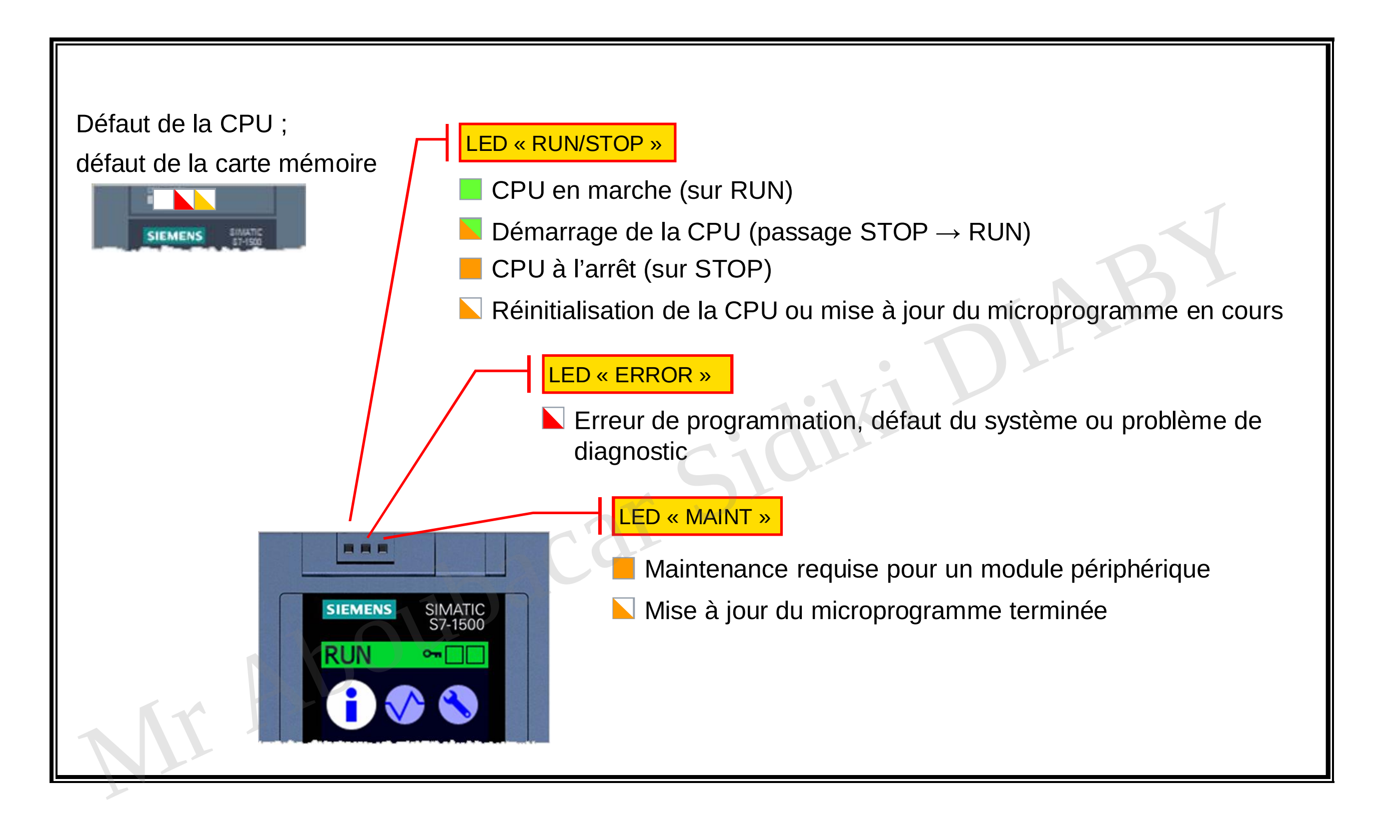
Le diagnostic système regroupe l'ensemble des fonctions de surveillance permettant d'assurer le fonctionnement correct de toutes les composantes du système d'automatisation. Toutes les CPU S7 sont dotés d'un système de diagnostic intelligent. Il n'est pas nécessaire de programmer la prise en compte des données de diagnostic par le diagnostic système. Elle est intégrée en standard au système d'exploitation de la CPU ou à d'autres modules dotés de la fonction de diagnostic et s'exécute automatiquement. Les CPU enregistrent les erreurs survenues dans la mémoire tampon de diagnostic et permettent aux personnels de maintenance d'effectuer un diagnostic d'erreur rapide et ciblé, même s'il s'agit d'erreurs sporadiques.

Réaction du système

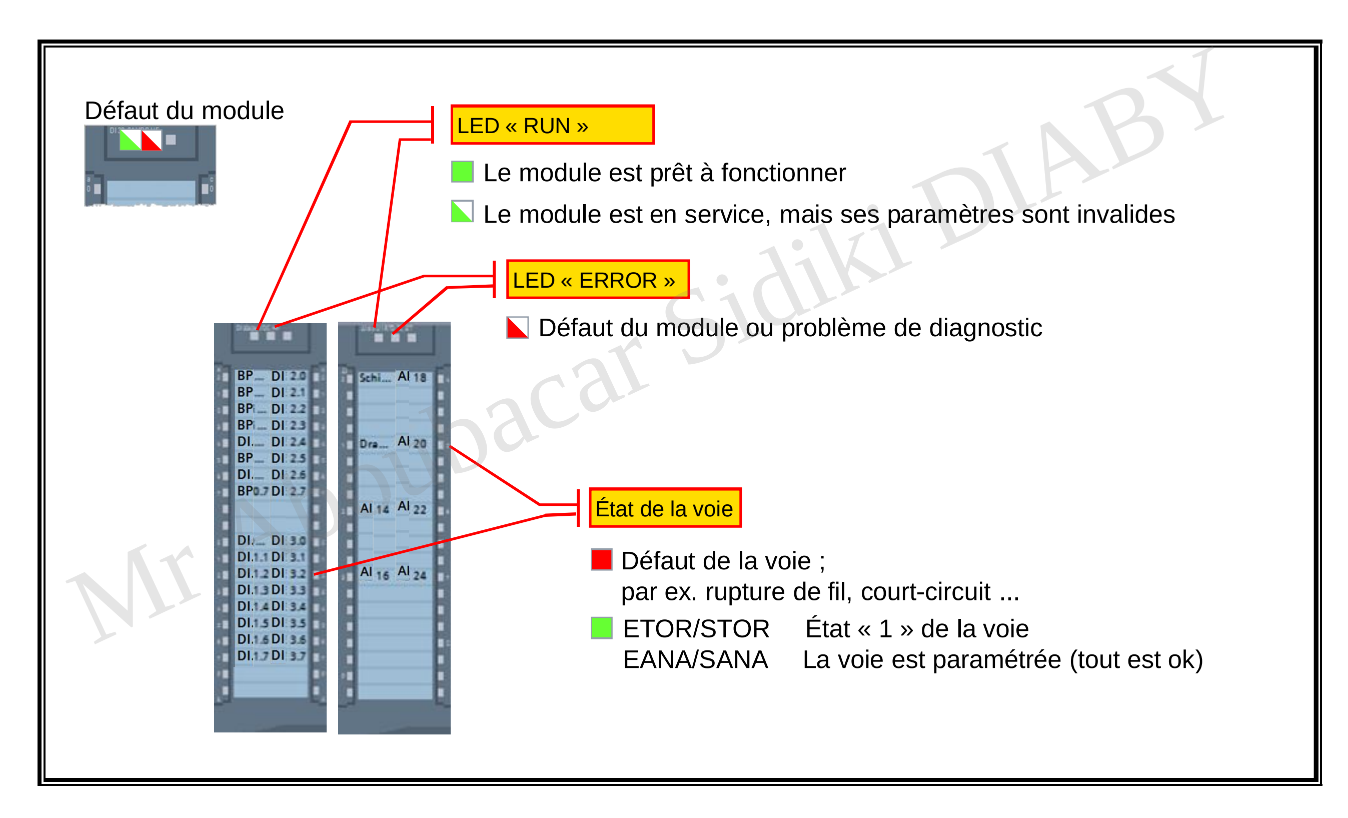
Lorsqu'il détecte une erreur ou un évènement bloquant, comme par ex. un changement d'état de fonctionnement (RUN à STOP), le système d'exploitation déclenche les actions suivantes:

- Inscription d'un message horodaté dans la mémoire tampon de diagnostic précisant la cause et les effets de l'erreur. Le tampon de diagnostic est une mémoire circulaire de la CPU destinée à enregistrer les évènements provoquant des erreurs. La taille du tampon de diagnostic est fonction de la CPU (par ex. CPU 315-2 DP = 100 entrées). Du fait de la structure circulaire de la mémoire, le dernier message de diagnostic écrase l'entrée la plus ancienne du tampon de diagnostic. La mémoire tampon de diagnostic n'est pas effacée, même après un effacement général de la CPU.
- Actualisation des listes d'état système, qui renseignent sur l'état des modules.
- Appel de l'OB d'erreur correspondant. Ce qui permet à l'utilisateur d'assurer sa propre gestion des erreurs.

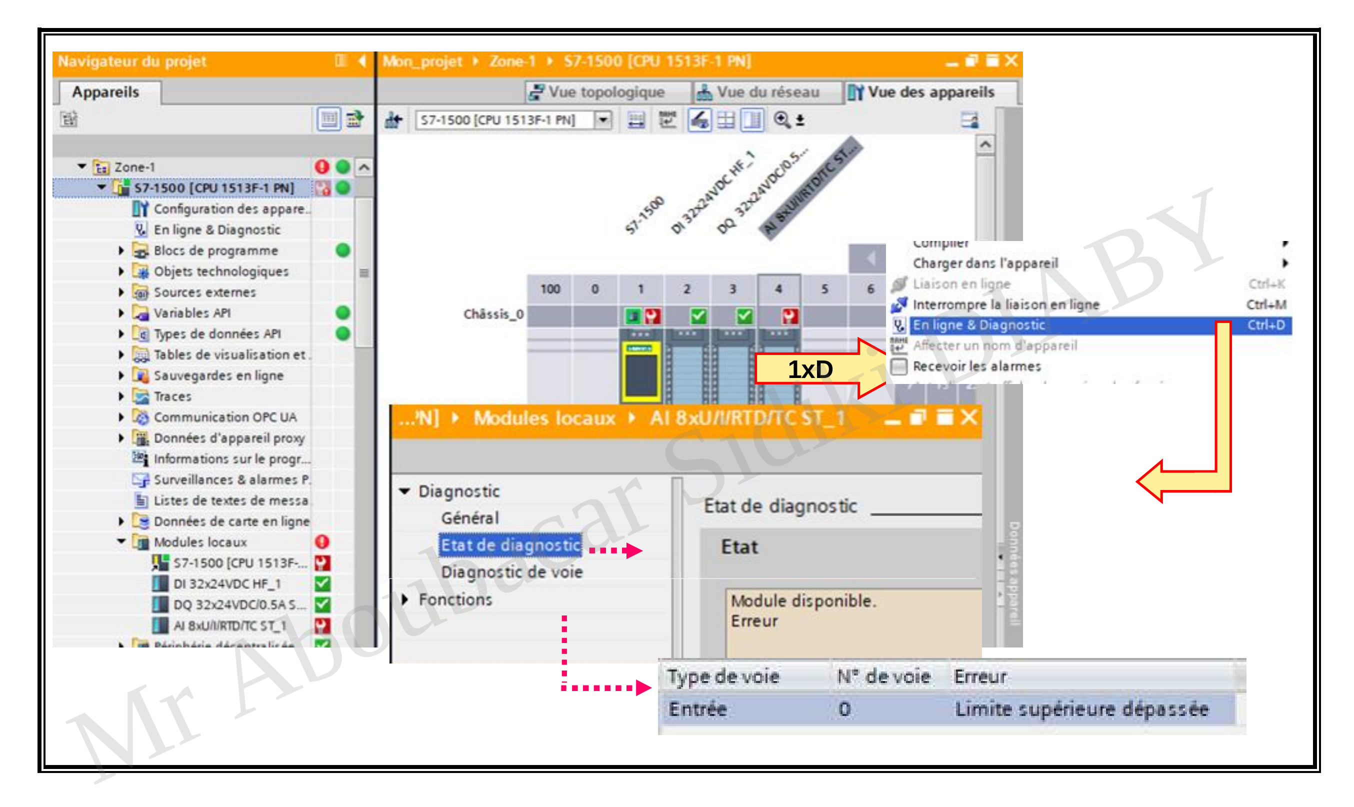
15.5.1. LED d'état sur la CPU du S7-1500



15.5.2. LED d'état sur les modules d'E/S centralisés du S7-1500



15.6. Diagnostic matériel



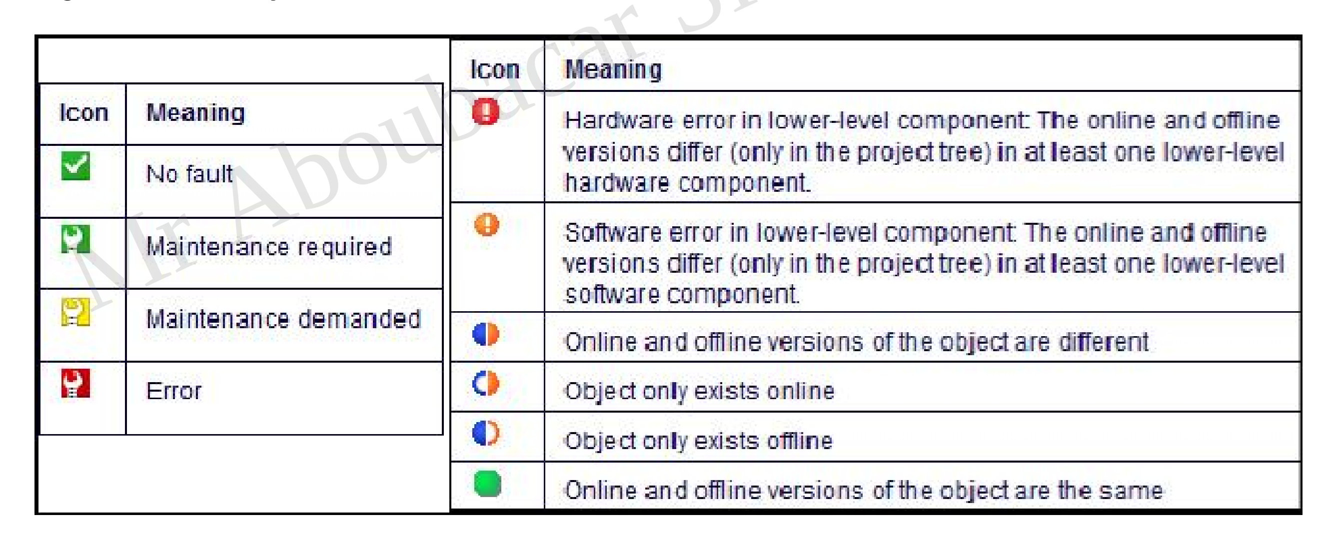
Diagnostic matériel

Pour utiliser cette fonction ouvrir la « configuration des appareils »et établir une liaison en ligne. La fonction de diagnostic matériel fournit des informations sur l'état ou l'état de fonctionnement des modules. La présence d'informations de diagnostic pour un module est signalée par des symboles qui indiquent l'état du module ou l'état de fonctionnement de la CPU.

Dans l'exemple ci-dessus, le module d'entrée analogique (Emplacement 7) a déclenché une alarme de diagnostic, ce qui entraîne le passage à l'arrêt de la CPU.

Les deux modules affichent les symboles correspondants. Par clic de souris droit sur la CPU puis « En ligne & Diagnostic » vous accédez au tampon de diagnostic de la CPU, pour le module analogique vous accédez aux données de diagnostic correspondantes (voir figure)

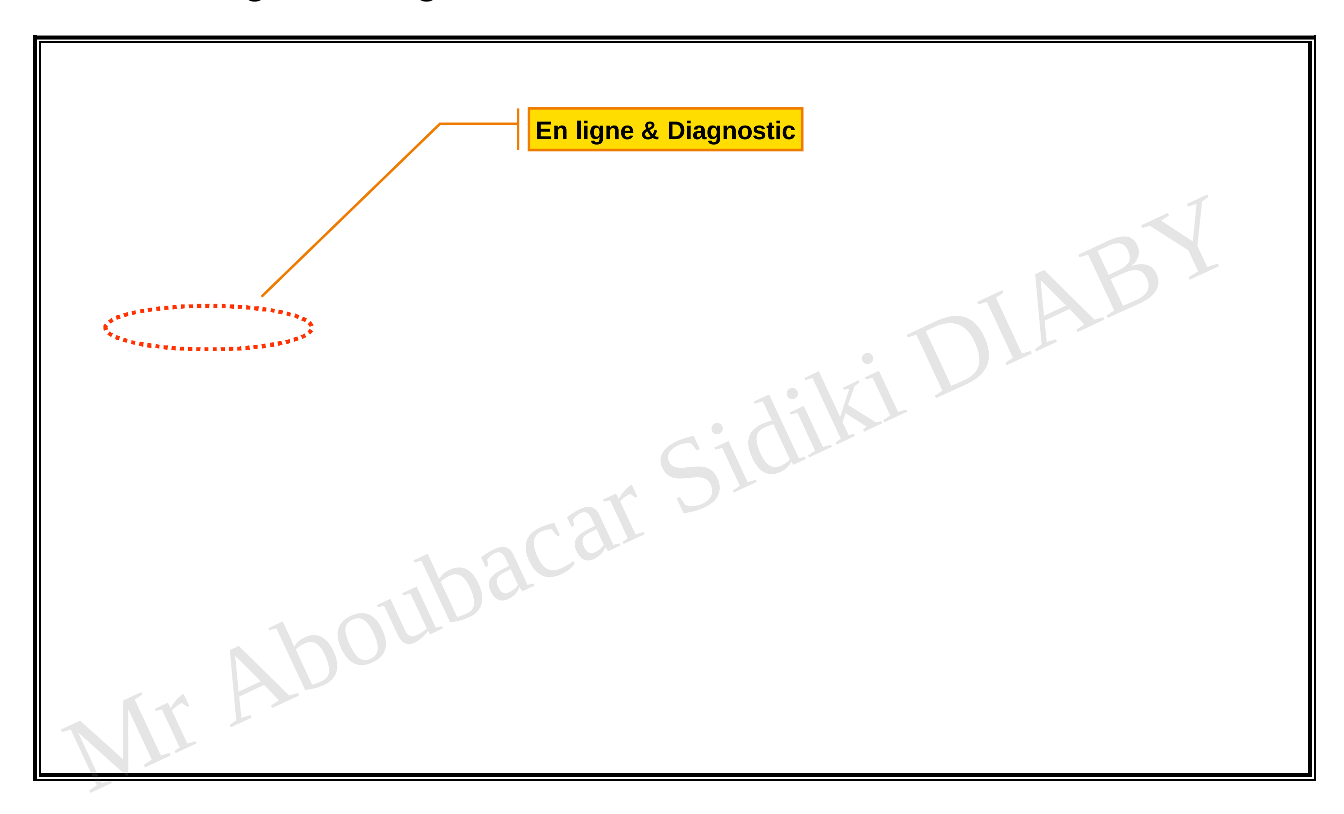
Signification des symboles



Icon	Meaning
	No fault
	Maintenance required
	Maintenance demanded
	Error

Icon	Meaning
	Hardware error in lower-level component: The online and offline versions differ (only in the project tree) in at least one lower-level hardware component.
	Software error in lower-level component: The online and offline versions differ (only in the project tree) in at least one lower-level software component.
	Online and offline versions of the object are different
	Object only exists online
	Object only exists offline
	Online and offline versions of the object are the same

15.7. En ligne & Diagnostic: Généralités

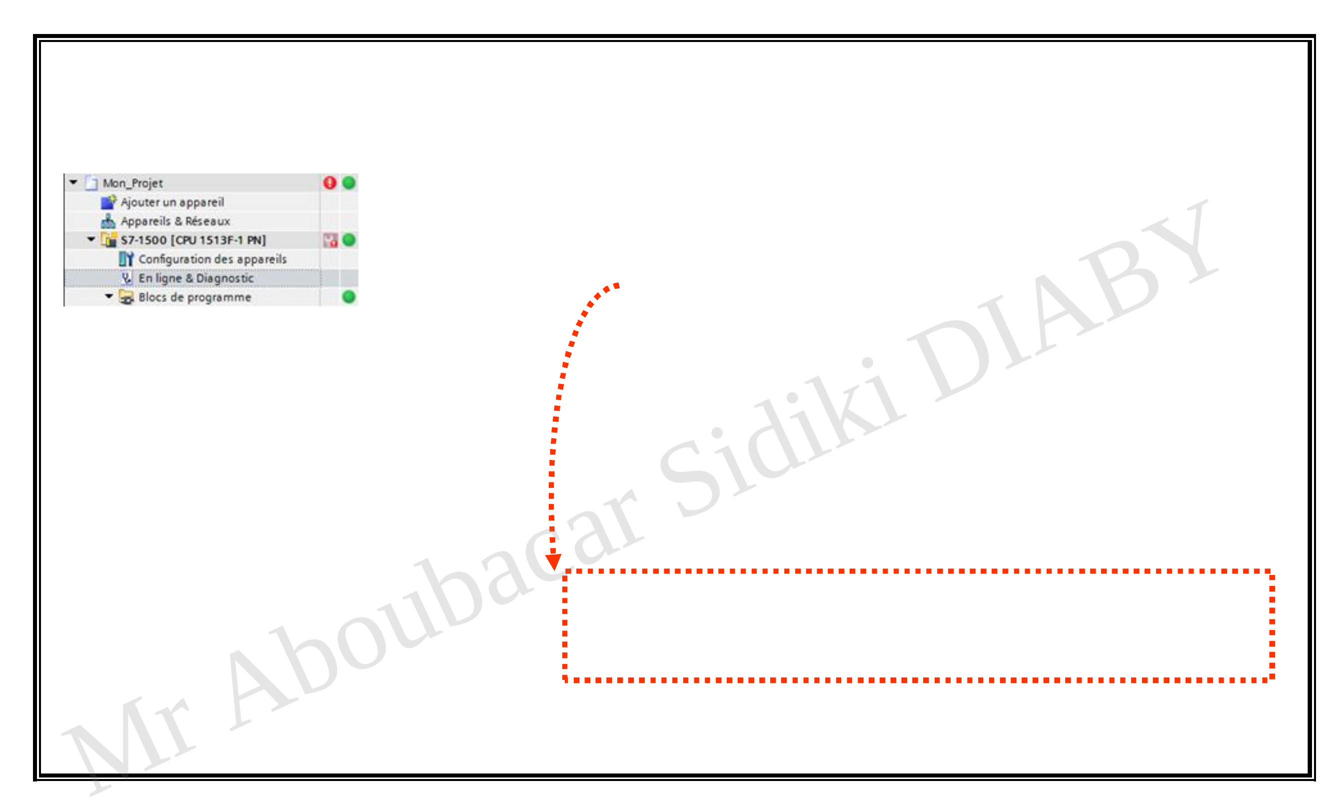


En ligne & Diagnostic

La fonction « état du module » lit les principales données du module directement raccordé. Chaque onglet délivre des informations complémentaire:

- Général: désignation du module, version du matériel et du firmware
- Tampon de diagnostic: il contient tous les éléments de diagnostic dans leur ordre d'apparition. Tous les évènements sont affichés en clair dans l'ordre chronologique de leur apparition.
- Mémoire: structure et affectation de la mémoire de chargement EPROM et RAM et de la mémoire de travail.
- Temps de cycle: affichage du temps de surveillance paramétré, de la durée du temps de cycle le plus court, le plus long et l'actuel
- Horloge système: affichage de l'horloge temps réel et du compteur horaire de fonctionnement intégré
- Caractéristiques: affichage des blocs systèmes intégrés et des blocs d'organisation disponibles ainsi que des zones d'opérandes (I, Q, M, T, C, L)
- Communication: Affichages des caractéristiques des interfaces de communication et vue d'ensemble des ressources de communication

15.7.1. En ligne & Diagnostic: Tampon de diagnostic CPU



Tampon de diagnostic

Le tampon de diagnostic est une zone mémoire sauvegardée de la CPU organisée en mémoire circulaire. Il contient tous les éléments de diagnostic (messages de défaut, alarmes de diagnostic, informations de démarrage etc.) de la CPU par ordre d'apparition. L'entrée la plus haute de la liste correspond à l'évènement le plus récent.

Tous ces évènements peuvent être affichés en clair sur la console de programmation dans l'ordre de leur apparition.



La taille du tampon de diagnostic est dépendante de la CPU. De même l'ensemble du tampon de diagnostic n'est pas sauvegardé en cas de coupure de tension (seule une partie est rémanente).

- Nombre d'entrées, 1000..3200
- Dont rémanentes 250...500

Détails sur l'évènement

Le champ « Détails sur l'évènement » contient des informations complémentaires sur l'évènement sélectionné:

- Désignation et numéro de l'évènement,
- Informations complémentaires, liées à l'évènement telle que par ex. l'adresse de l'instruction à l'origine de l'évènement etc.

Aide sur l'évènement

Ce bouton permet d'ouvrir l'aide sur l'évènement sélectionné dans la liste. (Dans notre exemple une erreur de programmation est survenue, mais l'OB d'erreur (OB121) n'a pas été programmé dans la CPU

15.7.2. Tampon de diagnostic de la CPU : Interprétation des messages d’erreur



Interprétation du tampon de diagnostic

Pour interpréter le tampon de diagnostic, les entrées se rapportant à un défaut doivent être examinées dans leur ordre d’apparition, donc du bas vers le haut:

Dans notre exemple, un démarrage à chaud a été effectué avant l’apparition du dernier défaut (Evènement N°4 et 5).

Entrées dans le tampon de diagnostic

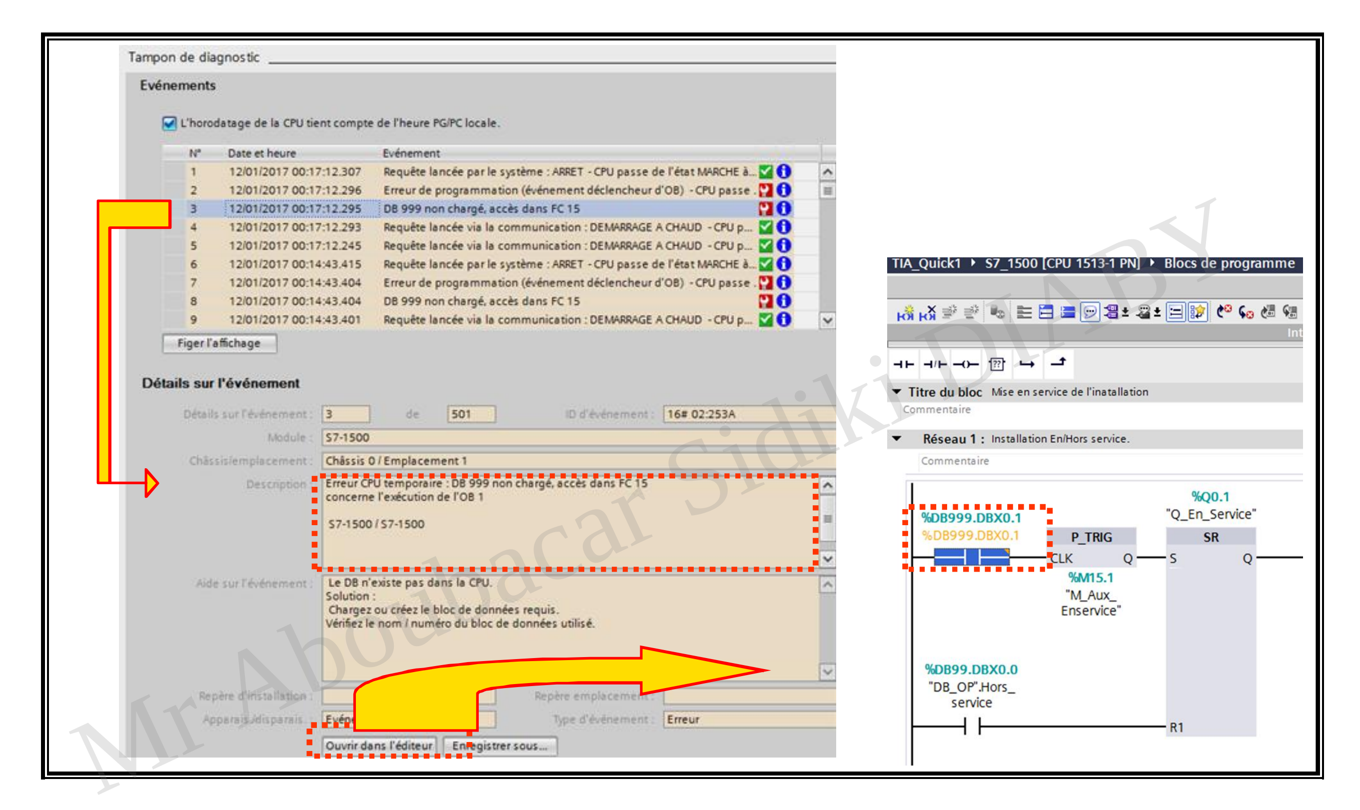
Le dernier défaut apparu après ce redémarrage a entraîné l’inscription des évènements suivants dans le tampon de diagnostic:

- <u>Evènement N° 3 :</u>
  Erreur de longueur de plage dans FC15

  <u>Détails:</u>
  - Concerne l’exécution OB1
    (FC15 est appelé dans le programme cyclique)
  - Accès en lecture d’une zone DB avec adresse incorrecte
    (l’accès à cette adresse spécifiée n’est pas réalisable)
- <u>Evènement N°. 2 :</u>
  Erreur de programmation (OB évènement de démarrage)
  (si présent, le système d’exploitation appelle un OB lors d’une erreur de programmation)

  <u>Détails :</u>
  - La CPU passe en STOP
  - Pas de traitement d’OB
    (l’OB 121 n’a pas été programmé)
- <u>Evènement N°.1</u>
  - La CPU passe en STOP

15.7.3. Tampon de diagnostic de la CPU : ouvrir le bloc erroné

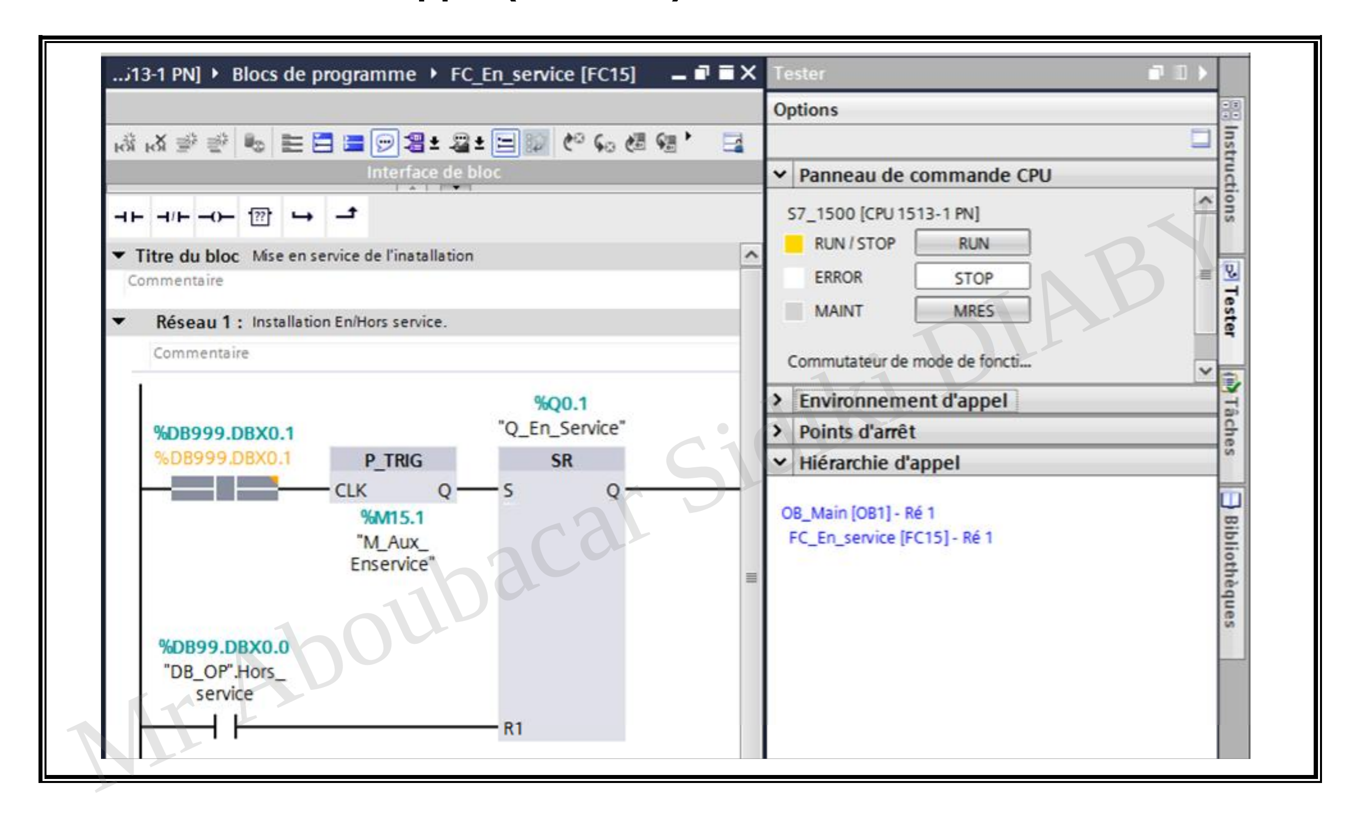


Ouvrir le bloc

Dans le cas d'erreurs synchrones, c'est-à-dire des erreurs déclenchées par une instruction erronée dans le programme utilisateur, le bouton « ouvrir dans l'éditeur » permet d'ouvrir le bloc dans lequel l'interruption est survenue.

En langage LIST, le curseur est placé directement devant l'instruction responsable de l'interruption. En CONT/LOG, le réseau concerné est sélectionné. L'exemple montre un accès en lecture au bit DB999.DBX0.1 qui n'existe pas dans la CPU.

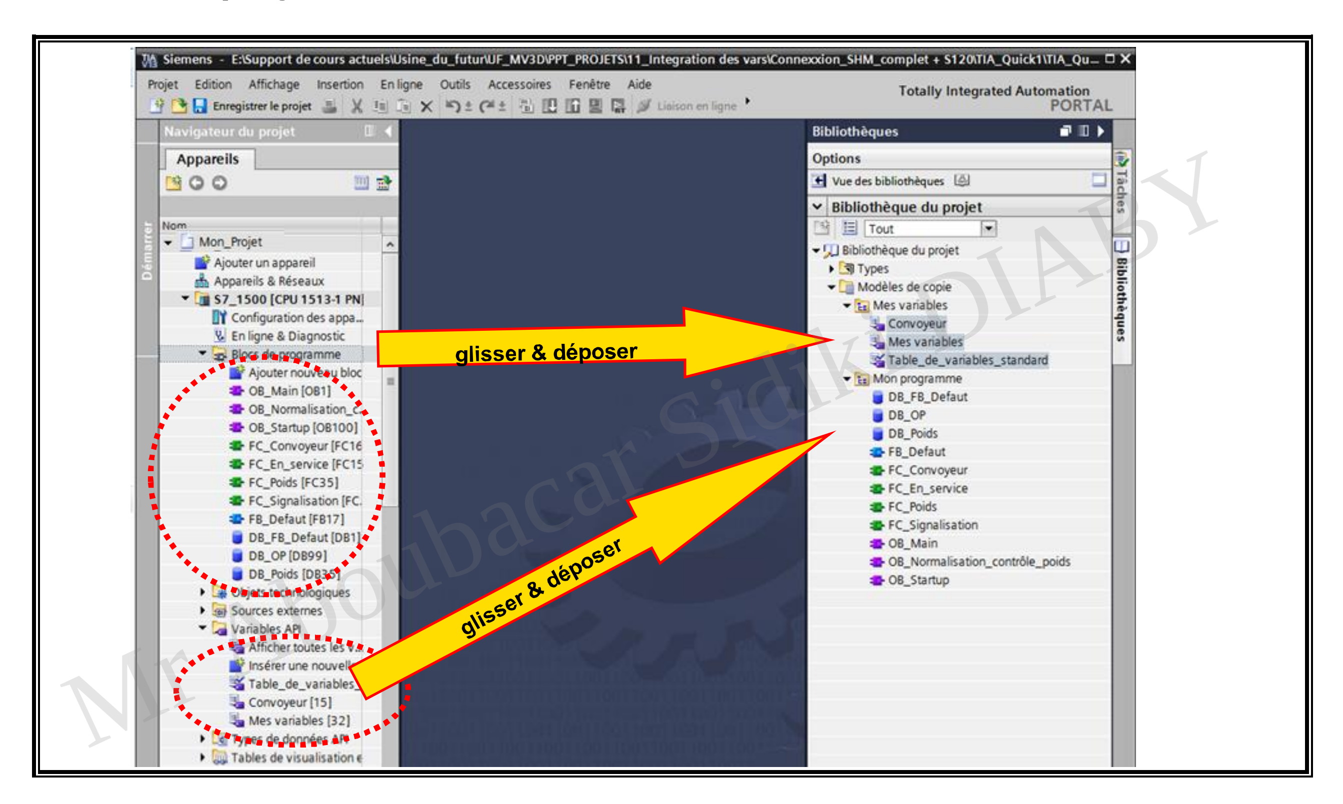
15.8. Hiérarchie d'appel (BSTACK)



Hiérarchie d'appel

Si des blocs paramétrables sont visualisés avec la fonction de test « visualisation de blocs » (état du bloc) et s'ils sont appelés plusieurs fois dans le programme alors « visualisation de blocs » ne fournit pas d'information univoque. Avec la fonction « Hiérarchie d'appel » il est possible de choisir l'appel du bloc dont l'exécution doit être visualisée, de sorte que la visualisation et l'état du bloc soient univoques.

15.9. Exercice 1: Créer une copie de sauvegarde dans la bibliothèque du projet



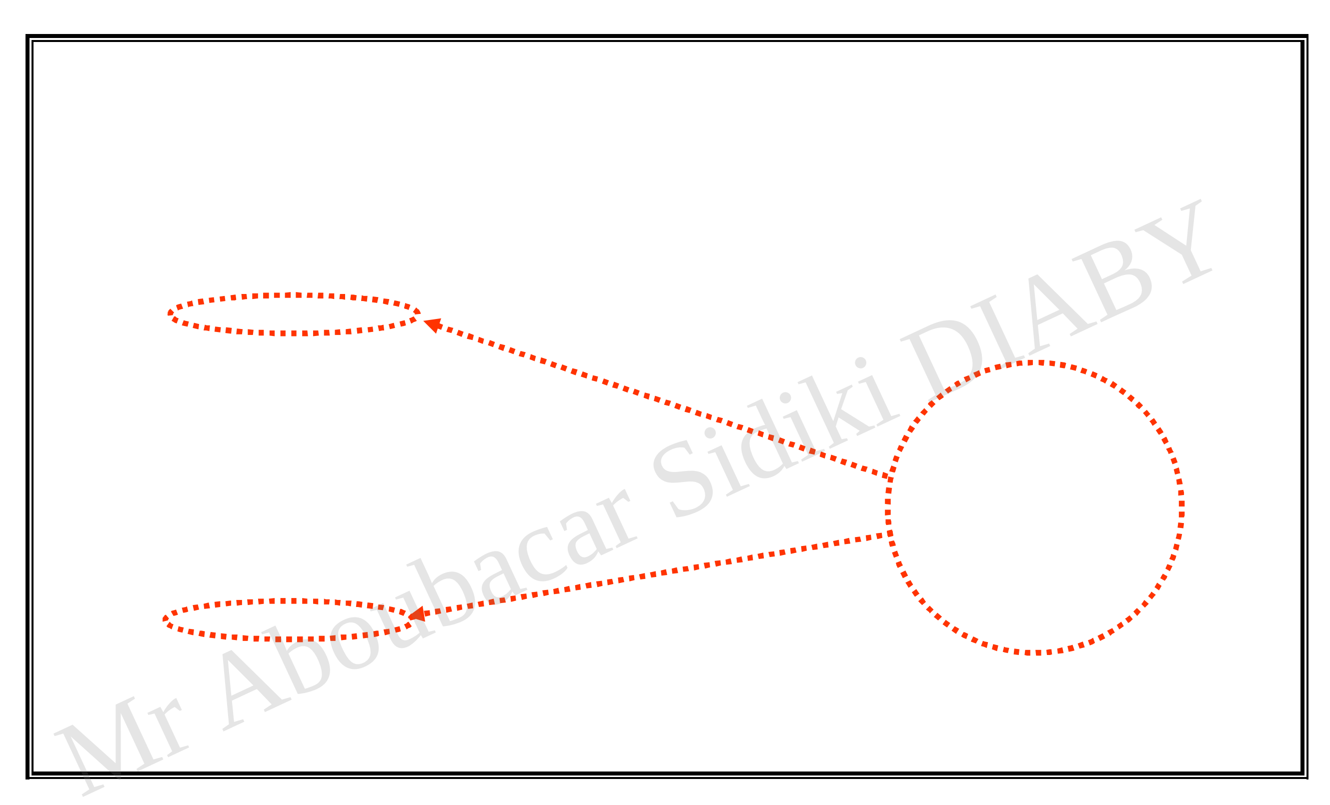
Enoncé

Créez une copie de sauvegarde de votre projet et des variables associées dans la bibliothèque du projet sous « Mes variables » « et « Mon_ programme », car vous allez travailler par la suite sur un programme erroné fourni.

Marche à suivre

1. Ouvrez « Bibliothèque du projet » dans les « Bibliothèques »
2. Sélectionnez tous les blocs S7 du programme de la CPU et copiez les par glisser-déposer dans votre bibliothèque du projet
3. Sélectionnez toutes les tables des variables et copiez les par glisser-déposer dans votre bibliothèque du projet
4. Sélectionnez votre IHM et copiez le par glisser-déposer dans votre bibliothèque du projet
5 Enregistrez votre projet avec la bibliothèque du projet.

15.9.1. Exercice 2: Copier le programme avec erreurs, les variables API et le TP700



Enoncé

Copiez dans votre projet les blocs ainsi que les variables « vars_prog_erreurs » ainsi que « Touchpanel » issus de la bibliothèque « Bib_Tia_PRO1».

Marche à suivre

1. . Puis effacer dans votre projet tous les blocs S7 et toutes les tables de variables API.
2 Ouvrez sous Bibliothèques globales la bibliothèque <Lecteur>:\exercices_TIA_Portal vxx\Bib_TIA_PRO1 »
3. Copiez par glisser-déposer les éléments de la bibliothèque suivant (voir figure).
4 Chargez tous les blocs du programme d'erreur dans la CPU en sélectionnant la CPU dans le navigateur de projet, puis en cliquant sur le bouton pour charger les blocs.
5. Redémarrez la CPU.
6. Sauvegardez votre projet.

15.9.2. Exercice 3:– Recherche d'erreur

Enoncé

Le programme d'erreur contient deux erreurs bloquantes , que vous allez corriger. L'exercice est terminé avec succès si après correction des erreurs et redémarrage de la CPU cette dernière reste en RUN.

En plus des erreurs reconnues par le système (Erreur de compilation ou STOP de la cpu) le programme contient encore une erreur de logique (Erreur RUN), ainsi la fonction correcte du programme n'est pas encore rétablie après résolution des erreurs bloquantes. Les erreurs logiques sont éliminées dans les exercices suivants.

Marche à suivre

Veuillez que chaque correction d'erreur STOP nécessite un redémarrage de la CPU. Si la CPU après ce redémarrage reste en STOP alors il subsiste encore une erreur STOP.

Lors de la correction, veuillez répondre aux questions suivantes au sujet des défauts rencontrés :

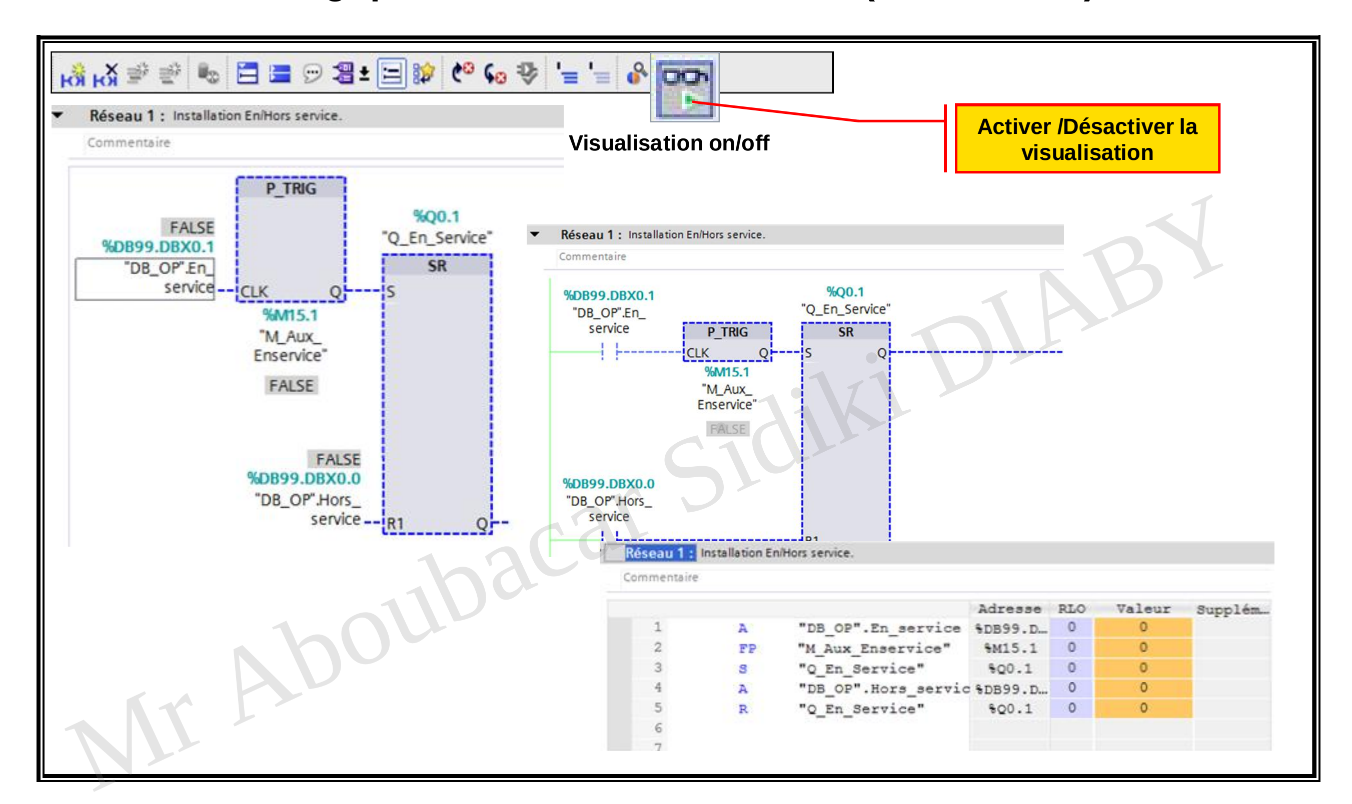
· Première erreur:

- Bloc interrompu: ……………………………………………………….........................….
- Erreur ....………………………………………..
- Correction: (ancienne instruction -> nouvelle instruction)
- ……….………………………………………………………....................……………...........……

· Deuxième erreur:

- Bloc interrompu: ....…………...………
- Erreur: ....……………………………..
- Correction: (ancienne instruction -> nouvelle instruction)
- ……………………………………………………………………………………………......................
..

15.10. Erreur logique: Visualisation d'un bloc (état du bloc)



Application

La fonction de test *Visualiser bloc* permet de suivre le traitement du programme au sein d'un bloc. L'état et le contenu des opérandes utilisés dans le bloc au moment du traitement du programme sont visualisés à l'écran. Le mode de test « Visualiser » (« Etat du bloc ») peut être activé pour le bloc ouvert dans l'éditeur CONT/LIST/LOG en cliquant sur le symbole des lunettes ou via la commande *Outils à Visualiser.*

Pour lancer la fonction de test, le bloc à contrôler peut être ouvert indifféremment *en ligne* ou *hors ligne*. Toutefois, si le bloc ouvert hors ligne ne correspondait pas au bloc enregistré en ligne dans la CPU, il faudra, au préalable, soit ouvrir le bloc enregistré en ligne soit charger le bloc ouvert offline dans la CPU et le visualiser par la suite.

En mode test, les états des opérandes ainsi que les éléments CONT/LOG s'affichent en différentes couleurs. Les réglages se réalisent via Outils à *Paramètres*:

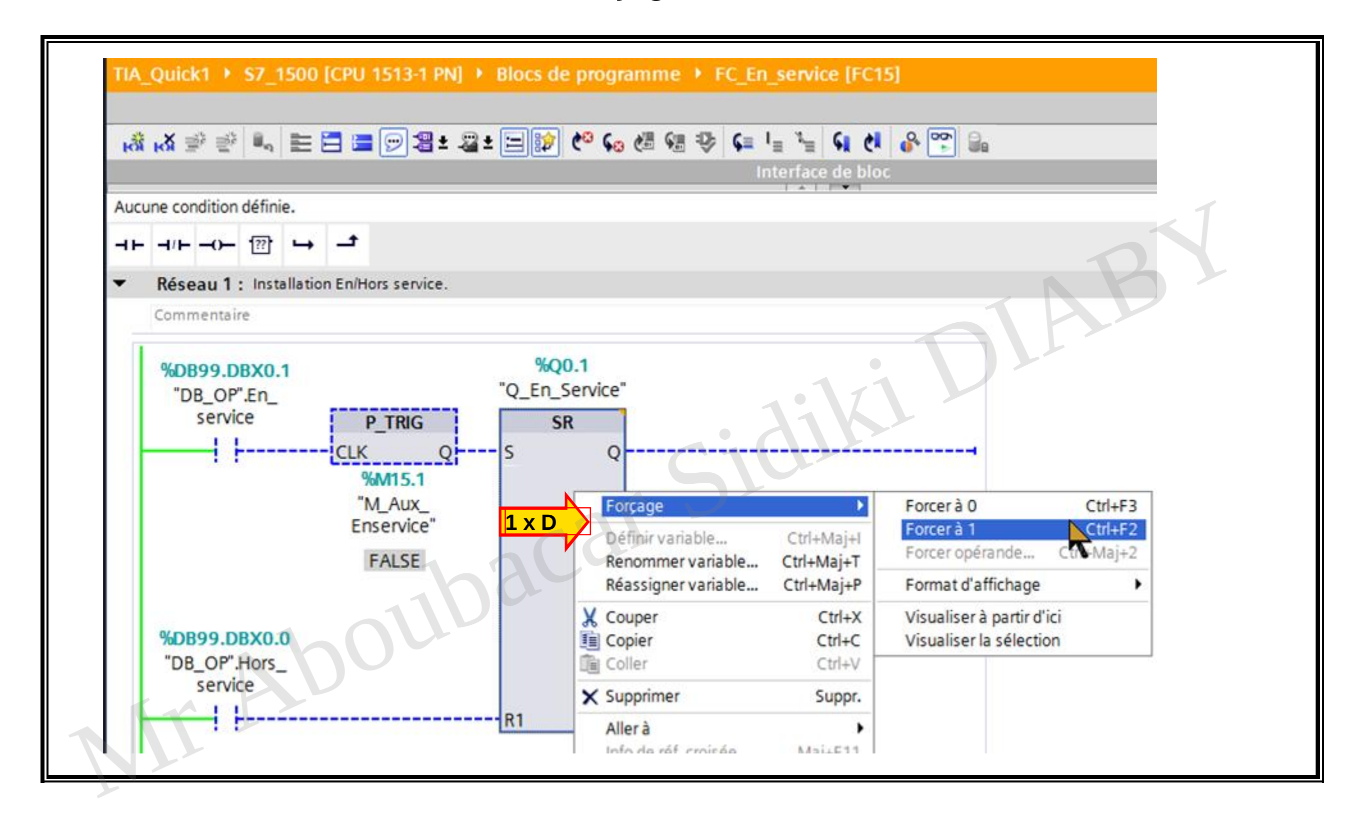
Exemples:

- Etat satisfait à « l'élément est représenté en vert »
- Etat non satisfait à « l'élément est représenté en bleu »

Remarque

L'affichage d'états n'est actif que si la CPU est en marche (RUN) et que les instructions à visualiser sont traitées !

15.10.1. Visualisation d’un bloc: Forçage

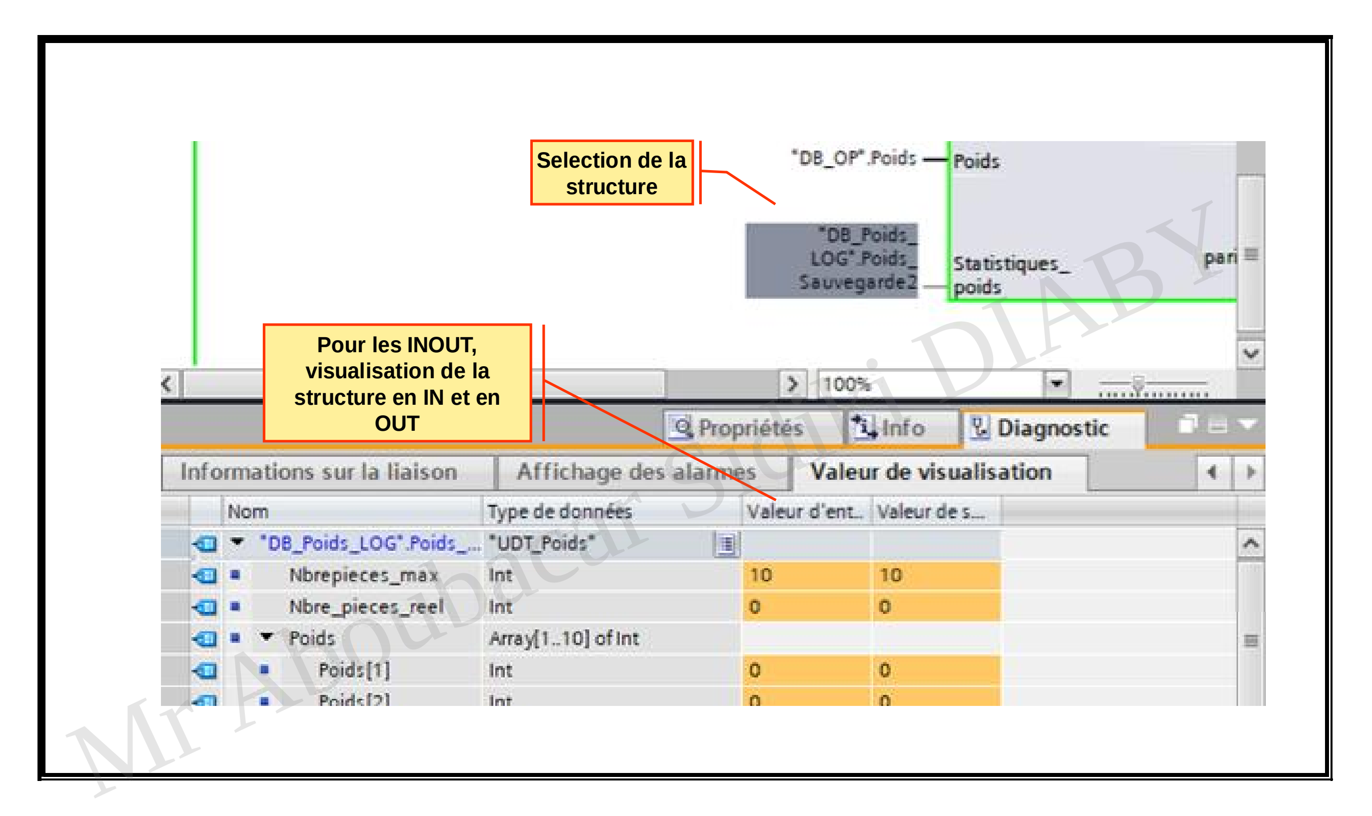


Forçage des variables

Lorsque la fonction de test « Visualiser un bloc »est active il est possible de forcer des variables à l’état « 0 » ou «1 ». L’affectation de l’état est unique.

Si la variable dont l’état a été forcé à « 0 » ou « 1 » n’est pas écrasée par le programme, elle reste dans son état. Si, par exemple, une sortie est pilotée à « 1 » et si cette variable n’est pas écrasée par le programme, alors la sortie reste activée ou à « 1 ».

15.10.2. Visualiser des structures



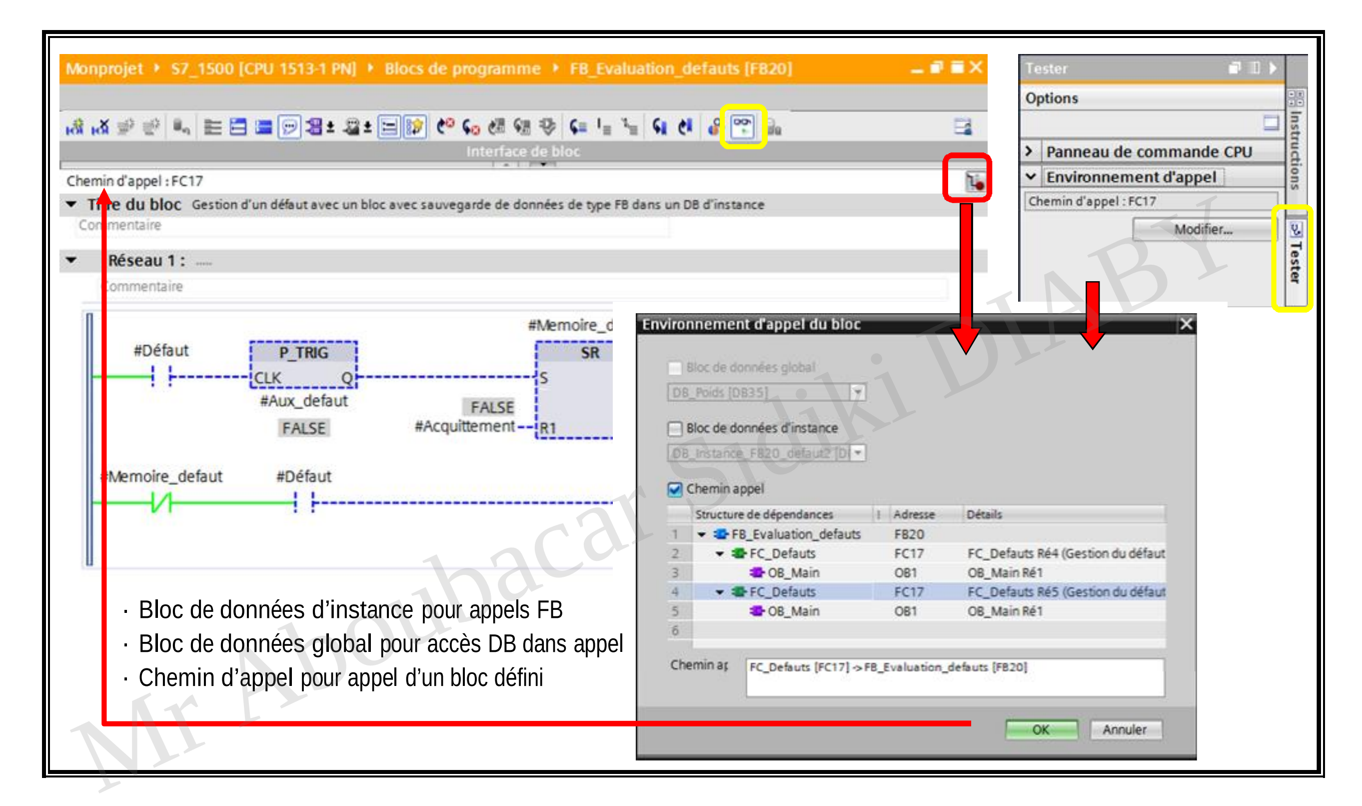
Règles de visualisation des structures (S7-1200/1500)

Lors de la visualisation de structures, les valeurs d'une variable API structurée sont visibles dans la fenêtre d'inspection à Diagnostic à Valeur de visualisation, avec les exceptions suivantes:

- Les structures, qui ont des propriétés de rémanence réglables ne peuvent pas être visualisées.

Pour afficher les valeurs pour une structure, il faut tout d'abord activer la visualisation via le menu contextuel.

15.10.3. Visualisation d'un bloc: Points de déclenchement

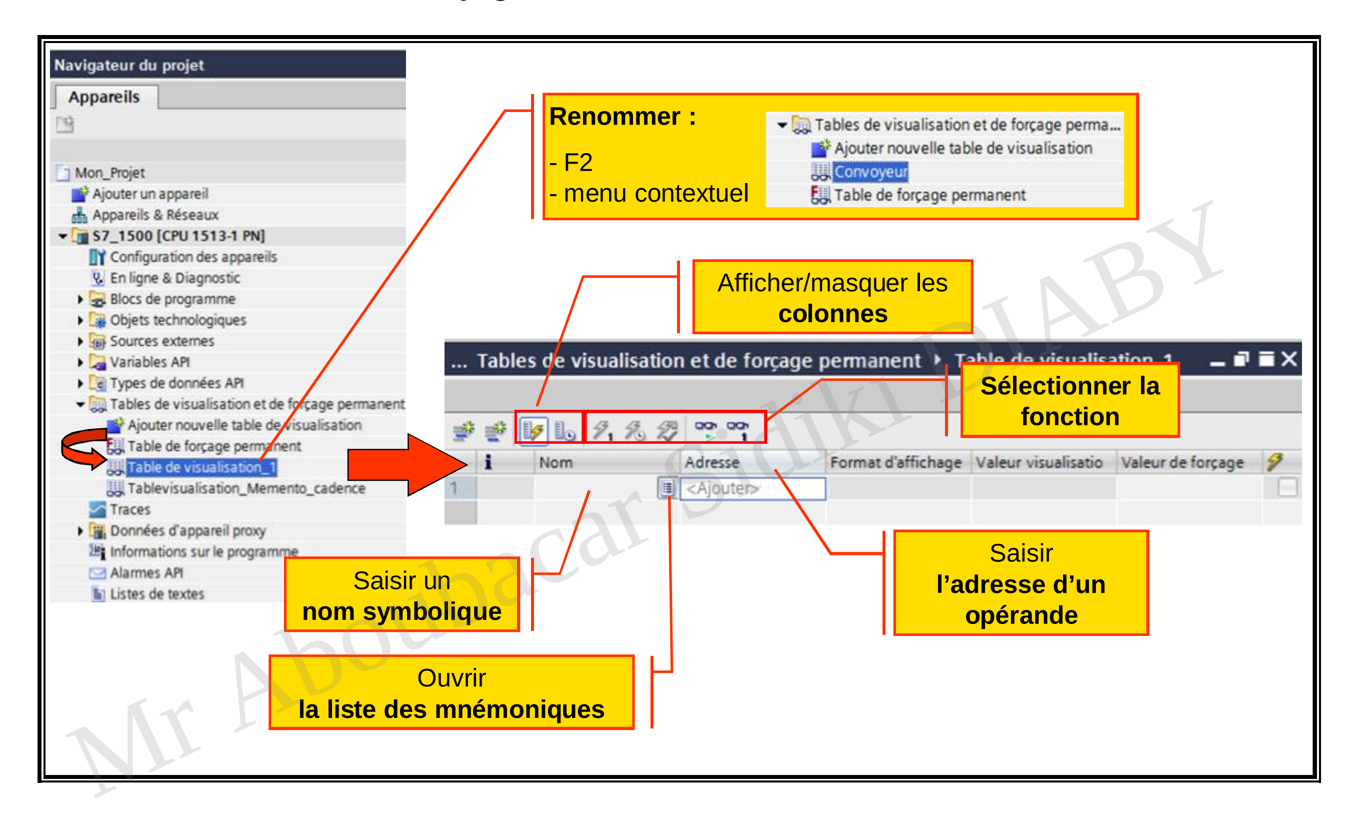


Fonction

Les conditions d'appel des blocs et des points d'arrêt sont paramétrables. Cela permet de définir les conditions dans lesquelles l'état d'un bloc doit être affiché ou l'exécution du programme doit être interrompue en un point d'arrêt. Les conditions suivantes peuvent être retenues:

- Bloc de donnée global
  - L'état d'un bloc ne s'affiche que lorsque le bloc est appelé avec le bloc de données global sélectionné.
- Bloc de données d'instance
  - L'état du bloc fonctionnel ne s'affiche que lorsque le bloc est appelé avec le bloc de données d'instance sélectionné.
- Chemin d'appel
  - L'état d'un bloc ne s'affiche que lorsque le bloc est appelé à partir d'un bloc défini ou d'un chemin défini.

15.10.4. Visualisation / forçage: Table de visualisation



Application

La fonction de test « Visualiser et forcer une variable » permet de visualiser et / ou de forcer des variables dans un format quelconque. Les variables souhaitées doivent être saisies dans une table de visualisation. A l'exception des variables locales et temporaires, toutes les variables et opérandes peuvent être visualisées et / ou forcées.

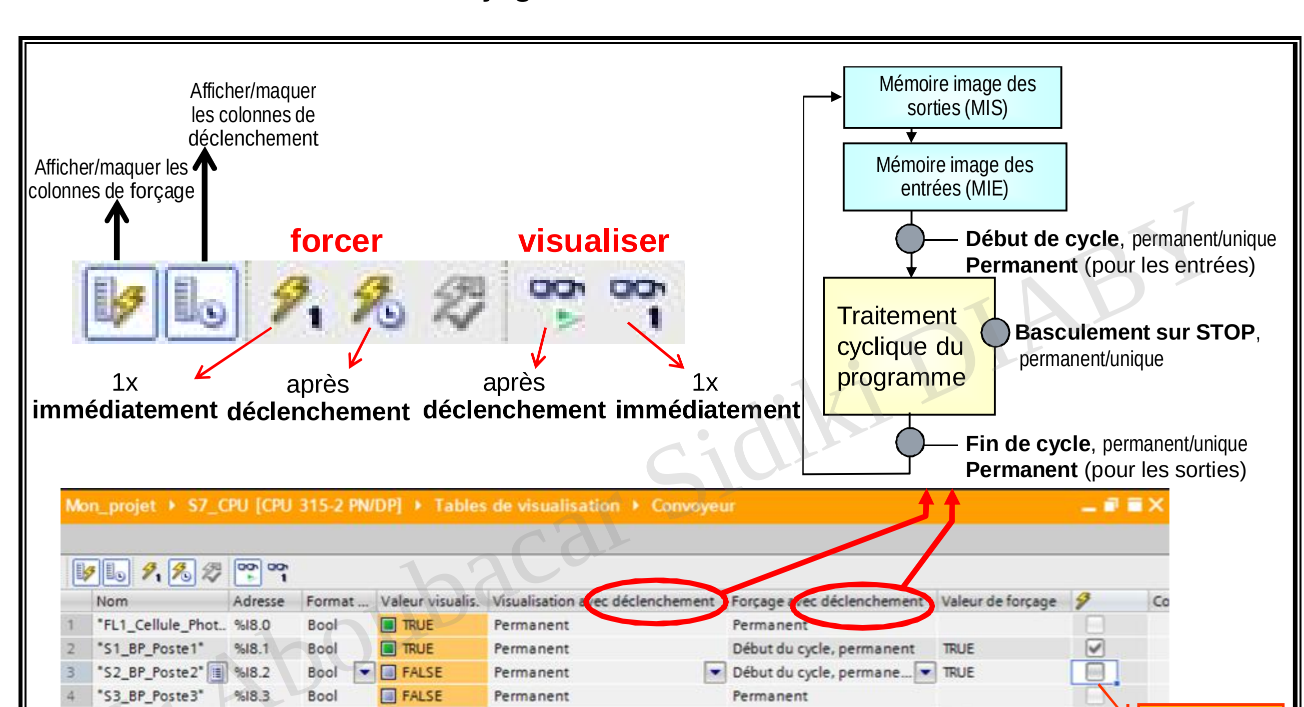
Les colonnes de la table de visualisation peuvent être sélectionnées via le menu *Affichage et ont la signification suivante*:

- **Nom:** nom symbolique des variables
- **Adresse**: adresse absolue des variables.
- **Commentaire:** Commentaire relatif à la variable affichée
- **Format d'affichage**: format d'affichage des données de la variable (par ex. binaire, décimal…) pouvant être sélectionné par un clic de souris
- **Valeur de visualisation**: valeur des variables dans le format choisi
- **Valeur de forçage**: valeur qui doit être affectée à la variable

Table de visualisation

Le nom de la table de visualisation peut être librement choisi. Les tables de visualisation enregistrées sont réutilisables pour la visualisation et le forçage : il n'est plus nécessaire de saisir à nouveau les variables à visualiser.

15.10.4.1. Visualisation / forçage de variables: Points de déclenchement



Points de déclenchement

Les colonnes « Visualisation ou Forçage avec déclenchement » permettent de définir les points de déclenchement pour *Visualiser et Forcer*. « Visualisation avec déclenchement » définit quand les valeurs des variables visualisées sont actualisées à l'écran. « Forçage avec déclenchement » définit quand les valeurs de forçage sont appliquées aux variables concernées.

Condition de déclenchement

« Condition de déclenchement pour visualisation » détermine s'il faut actualiser les variables une seule fois lorsque le point de déclenchement est atteint ou de manière cyclique (chaque fois que ce point est atteint).

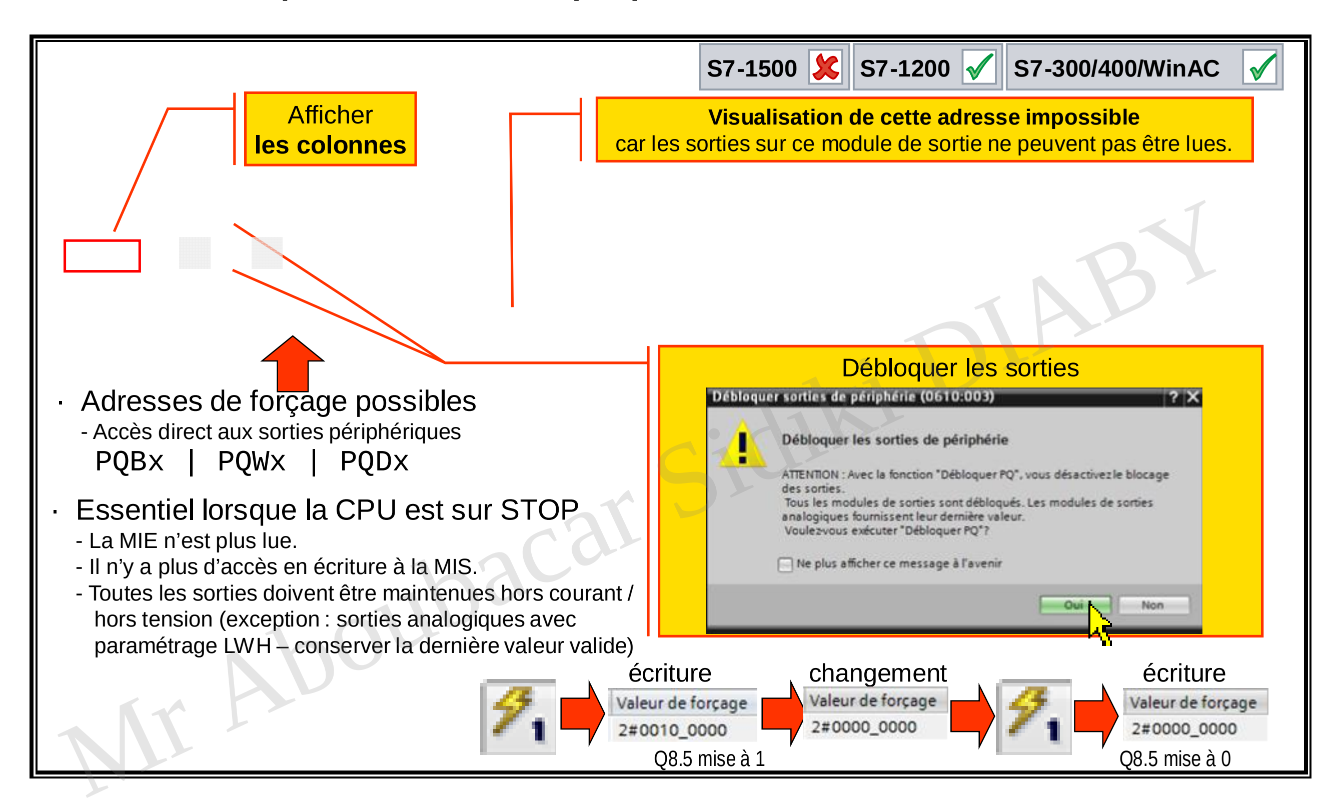
« Condition de déclenchement pour forçage » détermine s'il faut appliquer les nouvelles valeurs de forçage aux variables concernées une seule fois ou de manière cyclique (à chaque passage au point de déclenchement).

Application

En choisissant les points et les conditions de déclenchement appropriés, on peut réaliser entre autres les tests suivants:

- **Test de câblage des entrées***:* *Visualiser les variables, Point de déclenchement*: *début du cycle,* condition de déclenchement : *permanent*
- **Simuler l'état des entrées** (affecter un état, indépendamment du processus):
  Forcer des variables, Point de déclenchement: *début du cycle,* condition de déclenchement : *permanent*
- **Distinction défaut matériel / erreur logicielle** (un actionneur qui devrait être activé dans le processus n'est pas piloté)
  Visualiser les variables, pour visualiser la sortie correspondante, point de déclenchement: *fin du cycle, condition de déclenchement*: *permanent*
  (état de la sortie = « 1 » > logique du programme OK > Défaut processus (matériel)
  (état de la sortie = « 0 » > Erreur logique dans le programmer (par ex. double affectation)
- **Forcer des variables** (indépendamment de la logique du programme)
  Forcer des variables, point de déclenchement : fin de cycle, Condition de déclenchement: *permanent*

15.10.5. Débloquer des sorties de périphérie



Fonction « Débloquer les sorties de périphérie »

La fonction « Débloquer les sorties de périphérie » sert à vérifier l'accessibilité des modules de sorties, le câblage des modules de sorties TOR ou elle peut utilisée pour continuer à piloter des actionneurs du processus, même lorsque la CPU a été mise à l'arrêt (STOP) à la suite d'un défaut.

La fonction « Débloquer les sorties de périphérie » désactive l'inhibition des sorties de périphérie (PQ), ce qui permet le pilotage des sorties malgré la CPU en STOP.

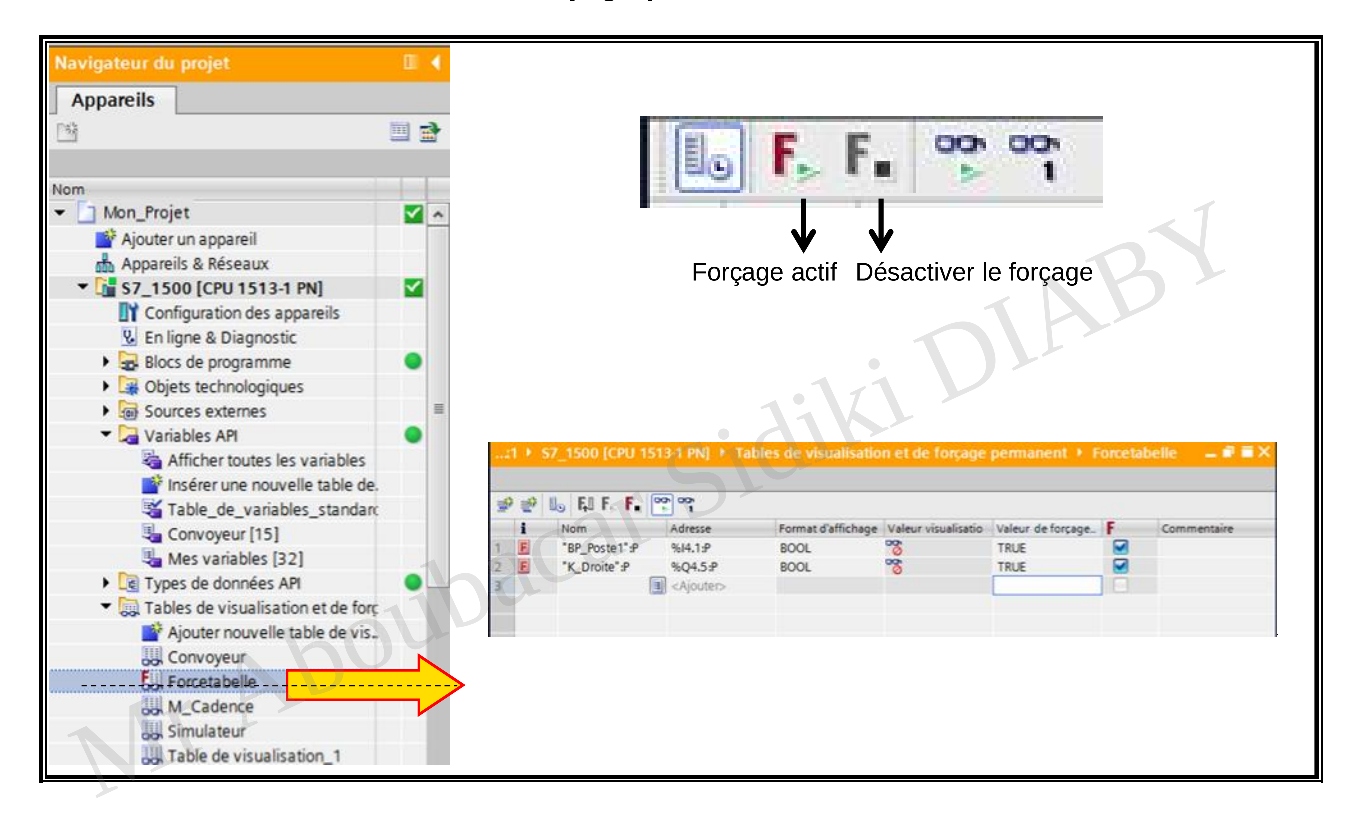
Condition

- La CPU doit se trouver à l'état STOP
- La CPU ne doit pas exécuter de contrat de forçage permanent
- La table de visualisation doit se trouver en « mode étendu », donc avec visualisation des colonnes avec déclenchement
- Les sorties périphériques à débloquer sont à saisir en octet, mot, double mot avec l'ajout **: P** (pour périphérie).
- Après le déblocage des sorties périphériques vous pouvez activer les valeurs de forçage via le bouton « Forçage unique » (pas via « Forçage avec déclenchement »).

Remarque

En cas de passage de la CPU de STOP vers RUN ou DEMARRAGE, « débloquer *PQ » est désactivé et un message s'affiche*.

15.10.6. Forcer des variables: forçage permanent



Fonction et utilisation

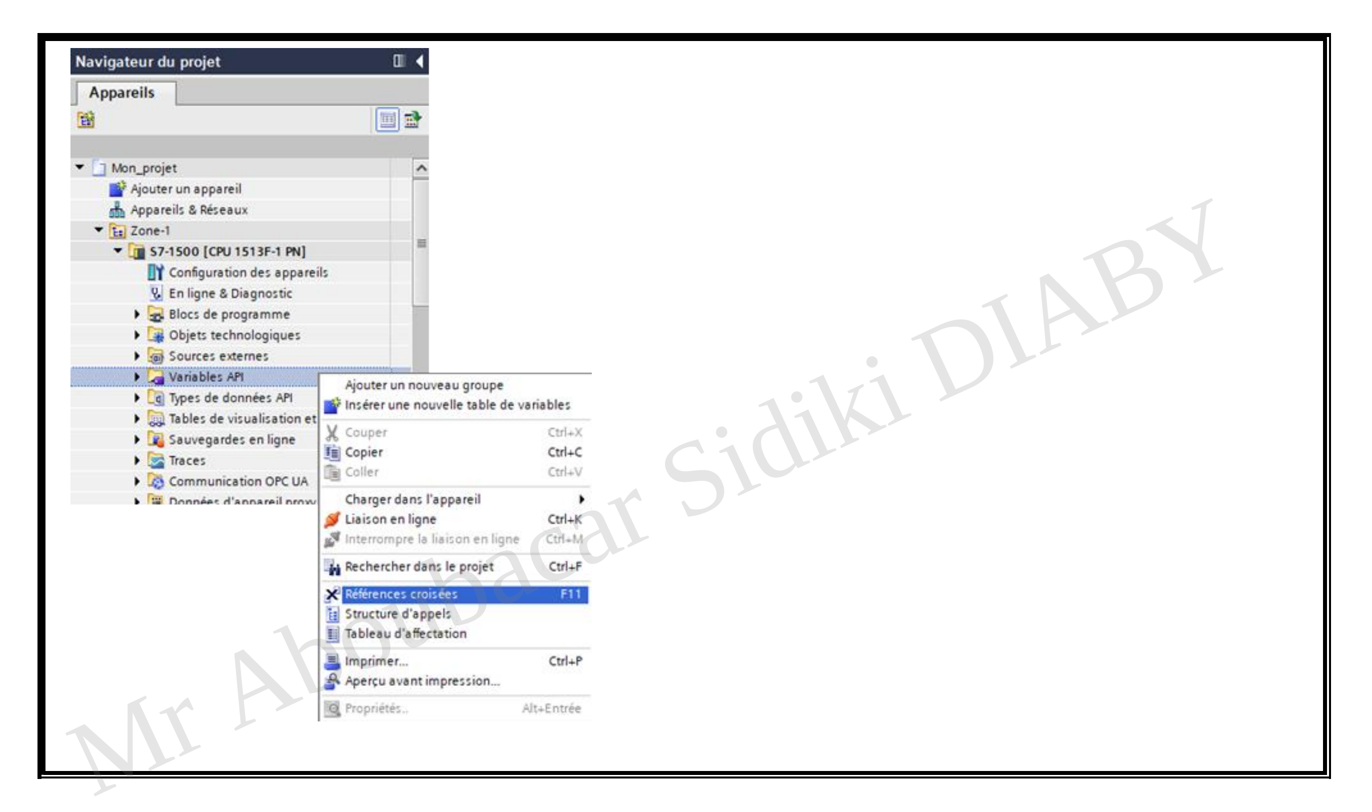
Le forçage permanent permet d'écraser des variables avec des valeurs quelconques indépendamment du programme utilisateur. Une seule fenêtre « Valeurs de forçage permanent » peut être ouverte par CPU.

Remarques sur le forçage permanent

Avant d'activer la fonction « Forçage permanent » assurez-vous que personne ne l'exécute simultanément sur la même CPU.

Une tâche de forçage ne peut être annulée <u>que par</u> un « Désactiver le forçage » explicite. (et non via *Edition à Annuler !).* La fermeture de la table de forçage ou le fait de quitter l'application <u>ne supprime pas</u> la tâche de forçage dans la CPU.

15.10.7. Données de référence: Références croisées des variables API



Introduction

La liste des références croisées donne une vue d’ensemble de l’utilisation des opérandes et des variables dans le programme utilisateur. A partir de cette liste de références croisées, il est possible de sauter directement à l’occurrence.

La liste de références croisées contient les informations suivantes :

- Quel opérande est utilisé dans quel bloc avec quelle opération,
- Quelle variable est utilisée dans quelle vue IHM,
- Quel bloc est appelé par quel autre bloc,
- En tant qu’élément de la documentation d’un projet, les références croisées fournissent une vue d’ensemble très complète de tous les opérandes, zone de mémoire, blocs, variables et vues utilisées.

Vues

Il existe deux vues de la liste des références croisées qui se distinguent par les objets respectivement affichés dans la première colonne :

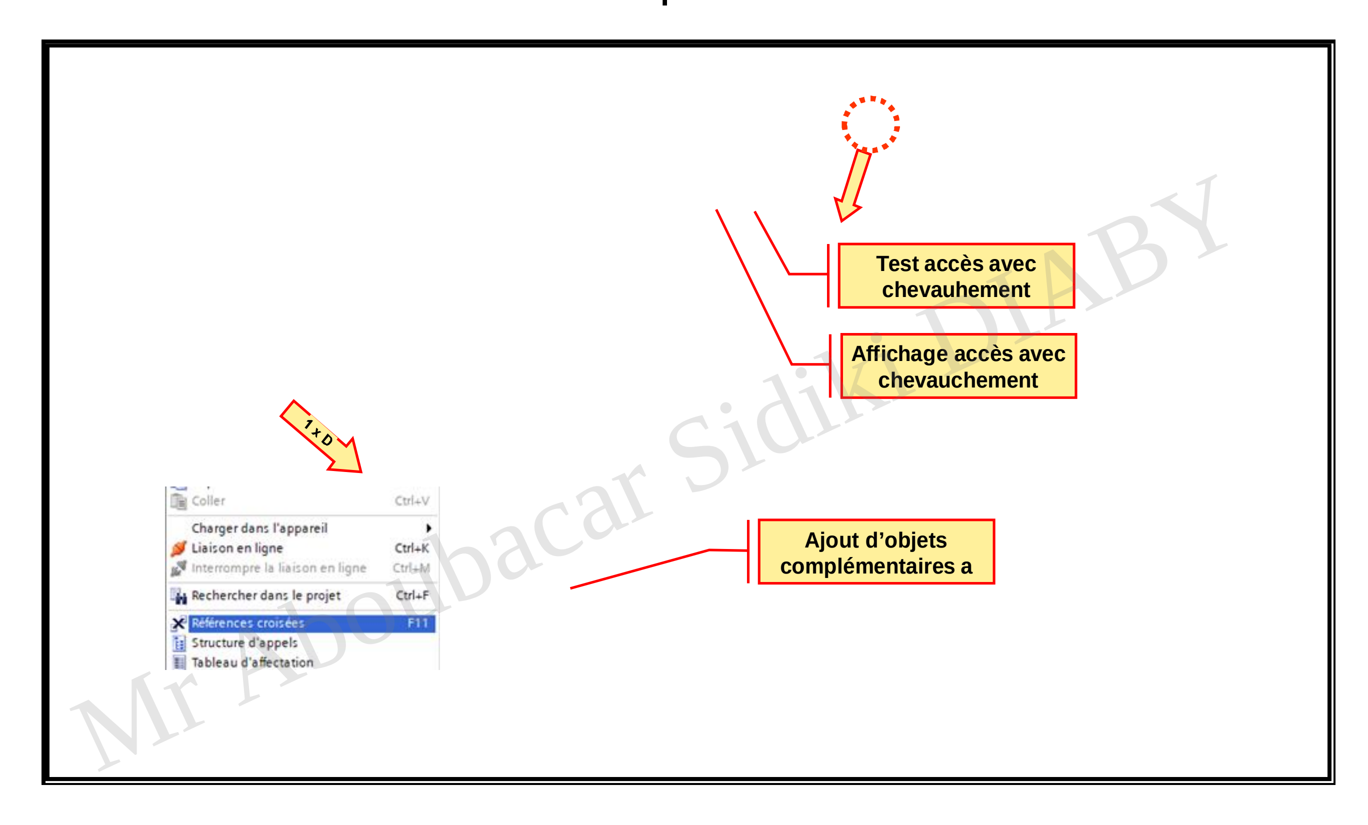
- Utilisé par :
  Affichage des objets référencés
  Affiche les emplacements où l’objet est utilisé.
- Utilisé :
  Affichage des objets de référencement.
  Affiche les utilisateurs de l’objet.

  Les info-bulles correspondantes fournissent des informations complémentaires pour les différents objets.

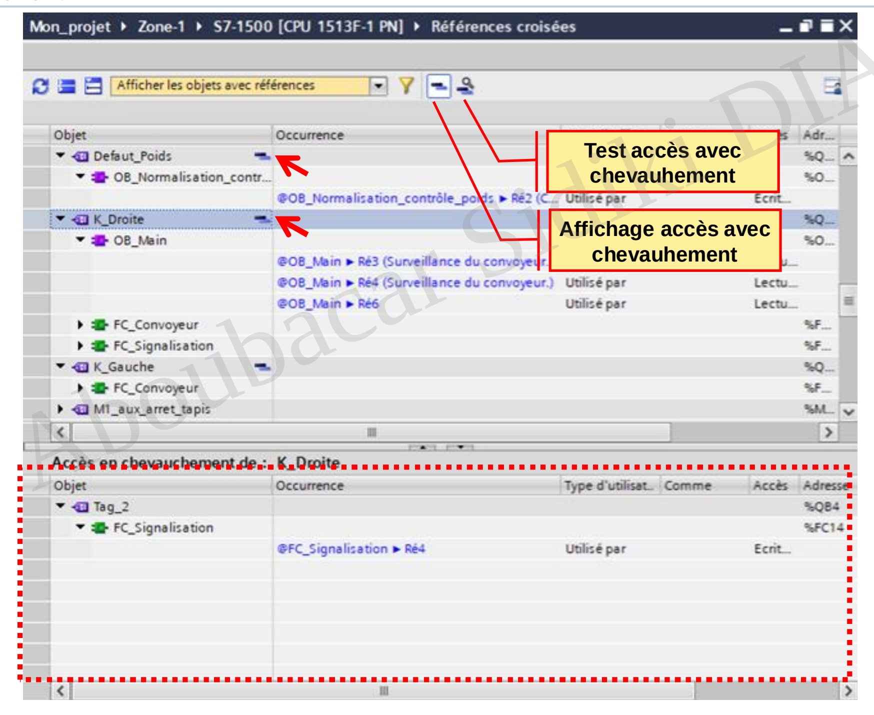
Afficher les variables non utilisées

Liste des variables déclarées dans la table des variables API mais qui ne sont pas utilisés dans le programme utilisateur S7.

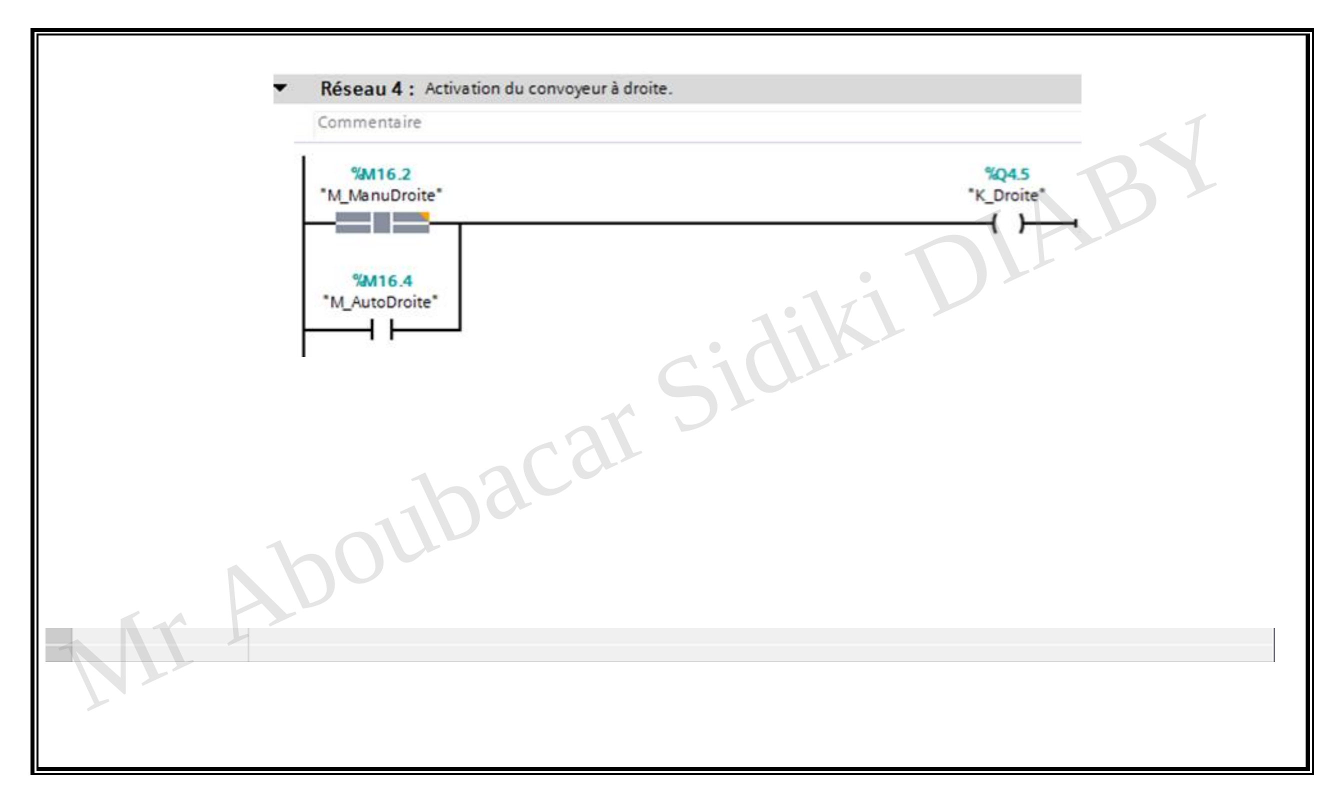
15.10.8. Données de référence: Accès qui se chevauchent



A l'aide du bouton « Vérifier les accès en chevauchement » vous pouvez contrôler l'existence d'accès en chevauchement pour une variable.
Dans ce cas ces derniers peuvent être affichés dans un tableau séparé par le bouton « Affichage des accès qui se chevauchent »



15.11. Données de référence: Références croisées d’une variable dans l’éditeur de blocs



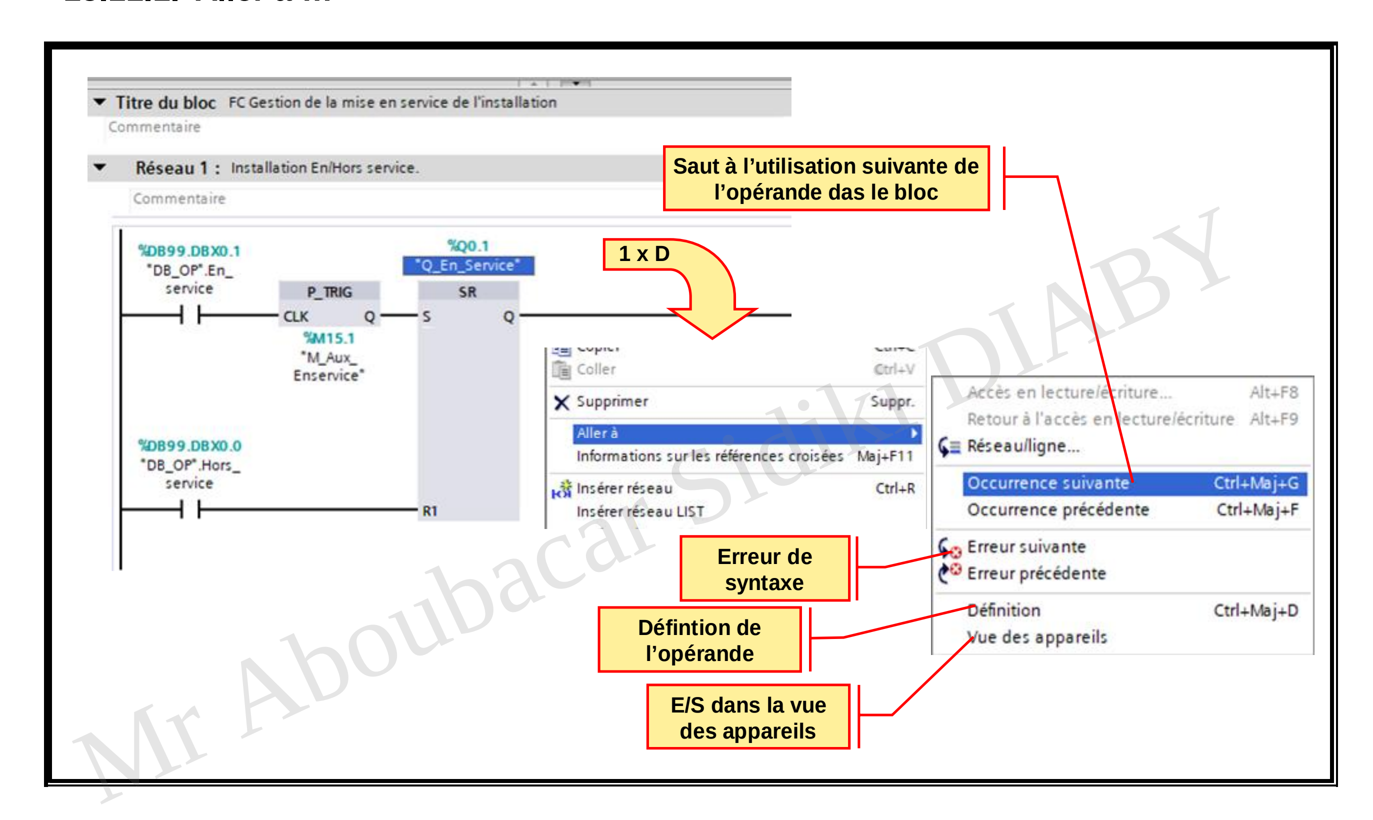
Introduction

Dans la fenêtre d’inspection, les informations sur les références croisées de l’objet sélectionné s’affichent dans l’onglet « Info > Références croisées ». Cet onglet indique, pour chaque objet sélectionné, à quels emplacements et par quels autres objets ce dernier est utilisé.
La fenêtre d’inspection affiche également les blocs des références croisées qui sont uniquement présents en ligne.

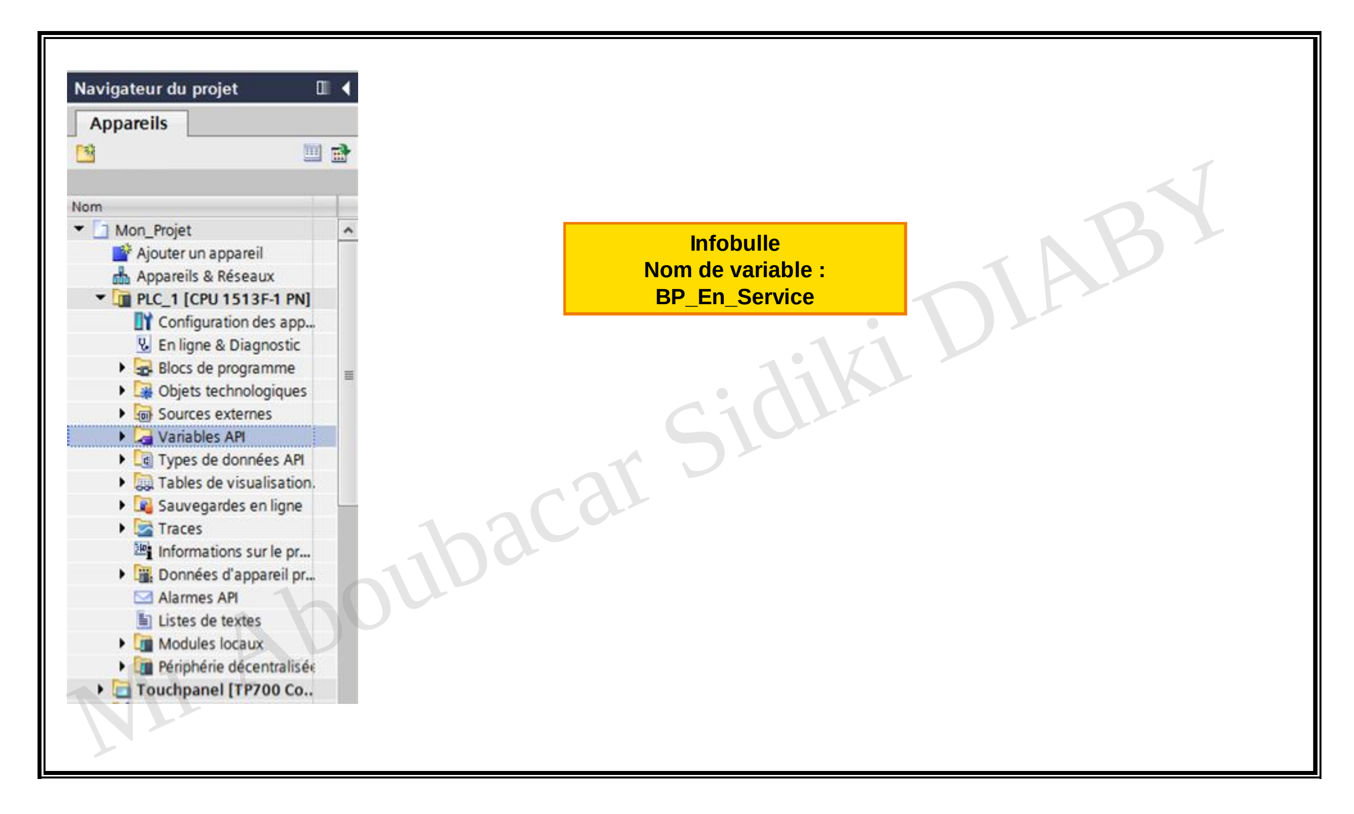
Structure

Les informations relatives aux références croisées sont représentées sous forme de tableau dans la fenêtre d’inspection. Chaque colonne contient des informations détaillées spécifiques au sujet de l’objet sélectionné et de son utilisation.

15.11.1. Aller à …



15.11.2. Données de référence: Affectation des opérandes I, Q, M, T, C (E, A, M, T, Z)



Affectation I/Q/M/T/C (E/A/M/T/Z)

Pour appeler le tableau d'affectation des I/Q/M/T/C, sélectionnez « Variables API -> Tableau d'affectation ».

Le tableau d'affectation donne une vue synoptique des bits et des octets dans les zones mémoire des entrées (I), des sorties (Q), des mémentos (M) et quelles temporisations et compteursS5 sont utilisés. Le mode d'accès (lecture ou écriture) n'est pas indiqué.

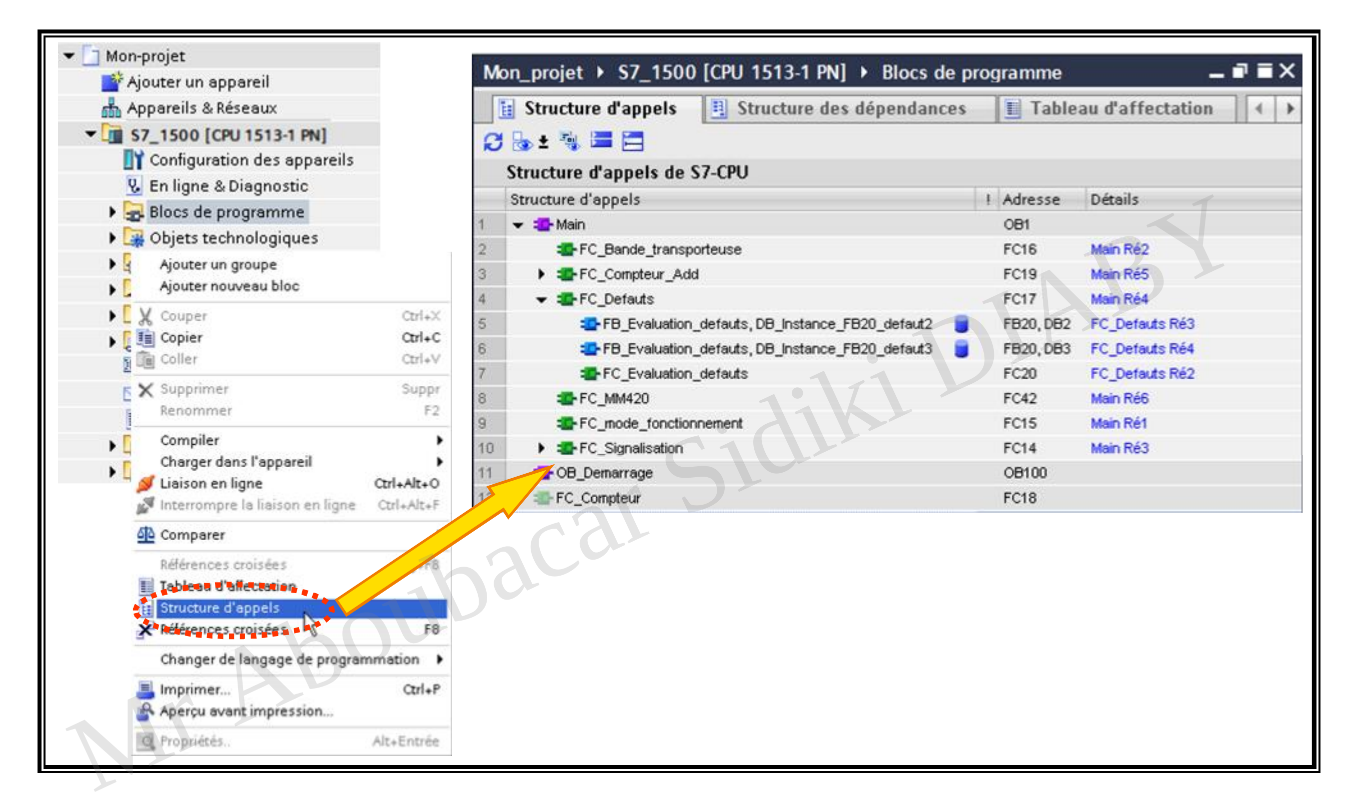
Les zones de mémoire pour les entrées (I), les sorties (Q) et les méméntos (M) sont affichées en lignes octet par octet.

Les bits repérés par un petit carré ou les opérandes binaires (par ex. I 1.0 ou M 17.0) sont utilisés de façon explicite dans le programme utilisateur.

Les octets repéres par un grand carré signalent des opérandes de format octet, mot ou double mot (par ex. MW20) utilisés dans le programme utilisateur. Le format des opérandes (octet, mot ou double mot) est indiqué par un trait vertical situé dans une des colonnes « B » (octet), « W » (mot) et « D » (double mot).

Les bits repérés par un petit et un grand carré, sont utilisés de façon explicite dans le programme utilisateur en tant qu'opérandes binaires et via un opérande au format octet, mot ou double mot.

15.11.3. Données de référence: Structure des appels

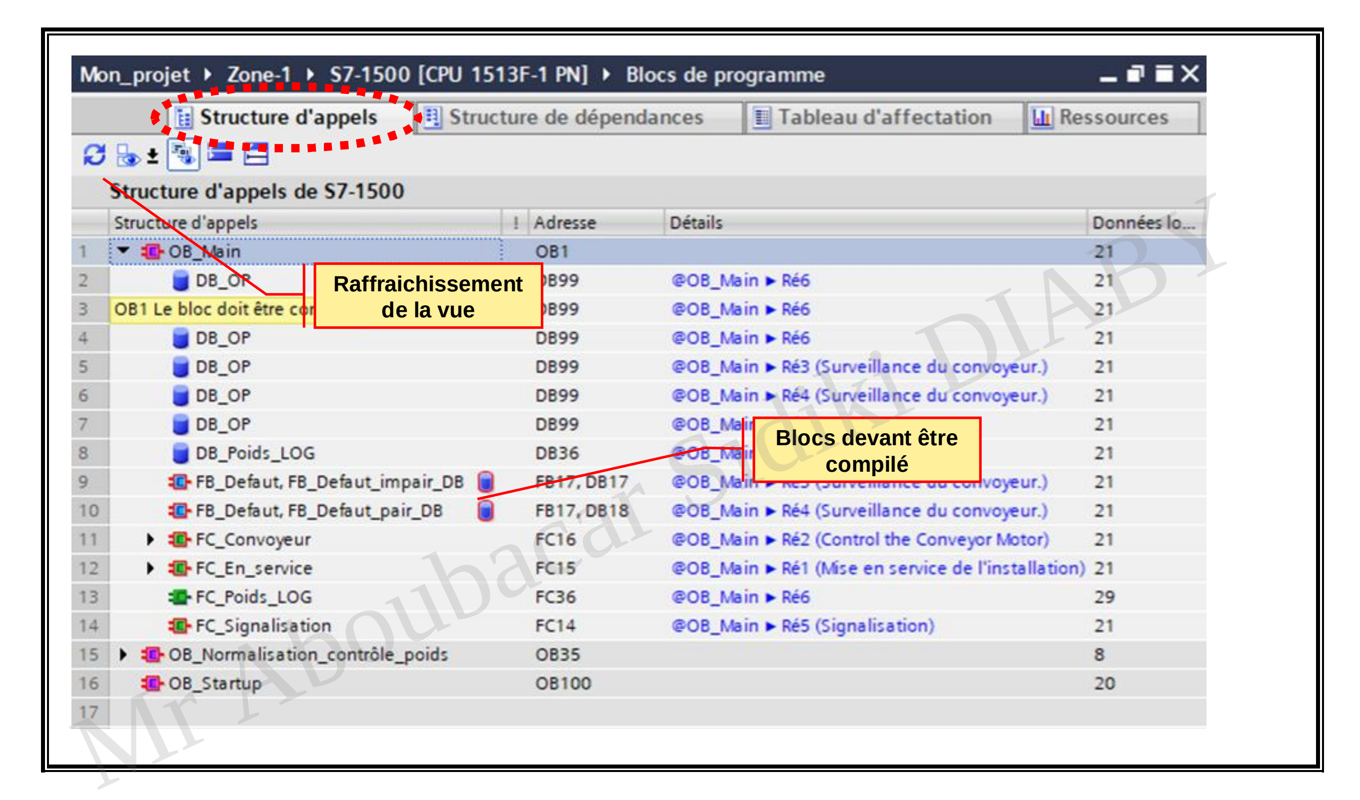


Structure des appels

La structure des appels décrit la hiérarchie des appels des blocs au sein d'un programme S7.Elle donne une vue d'ensemble sur :

- Les blocs utilisés
- Les sauts aux emplacements où les blocs sont utilisés
- Les relations de dépendance entre les blocs
- Les données locales nécessaires aux blocs
- L'état des blocs

15.11.4. Données de référence: Structure des dépendances

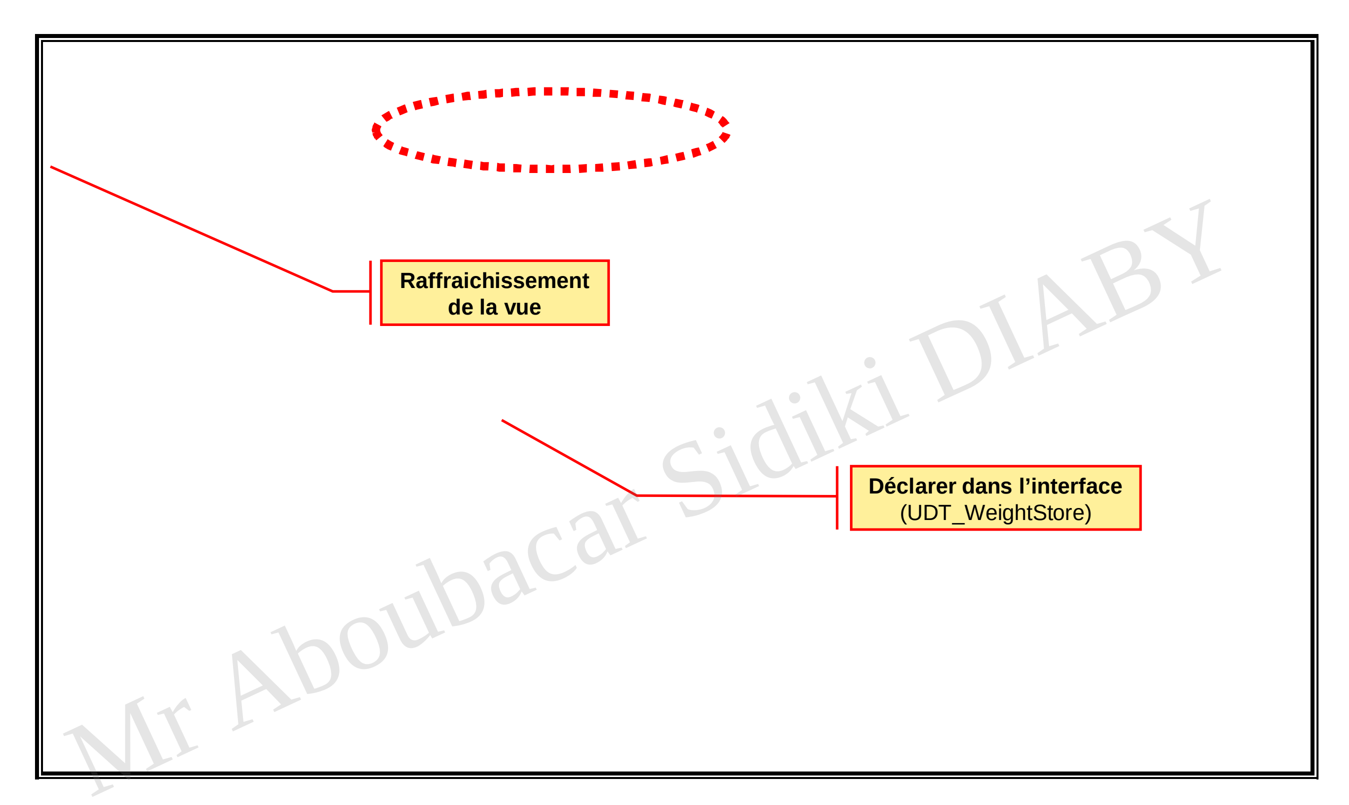


Structure de dépendance

L’affichage de la structure de dépendance vous donne une liste des blocs utilisés dans le programme utilisateur. Tout à gauche se trouve le bloc concerné sous lequel apparaissent, en retrait, les blocs appelant ou utilisant ce dernier.

La structure des dépendances signale en outre l’état des différents blocs par des symboles. Les objets créant un conflit d’horodatage qui peuvent donc entraîner une incohérence dans le programme sont signalés par différents symboles.
La structure de dépendance constitue une extension de la liste de références croisées pour les objets.

15.11.5. Données de référence: Structure des dépendances

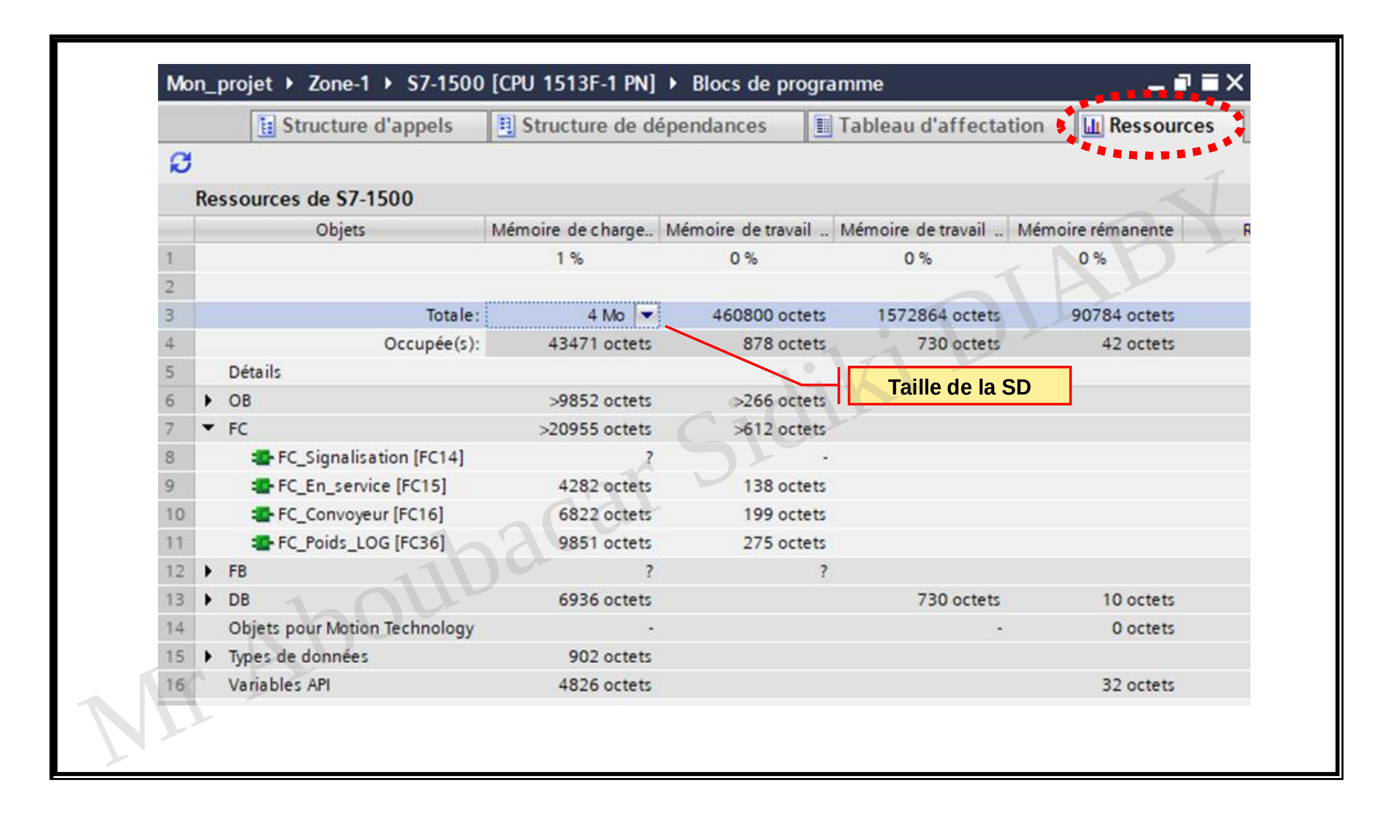


Structure des dépendances

La structure des dépendances s'ouvre via le menu « Accessoires > Structure des dépendances » et affiche la liste des blocs utilisés dans le programme utilisateur. Dans un premier niveau (le plus à gauche) vous retrouvez respectivement le bloc puis en dessous et en indentation les blocs qui appellent ou utilisent ce bloc.

La structure des dépendances affiche également l'état des blocs individuellement par l'utilisation de symboles. Les objets qui causent un conflit d'horodatage qui peut amener une incohérence dans le programme sont identifiés avec différents symboles.
La structure des dépendances représente une extension de la liste des références croisées pour objets.

15.11.6. Ressources



	Objets	Mémoire de charge..	Mémoire de travail ..	Mémoire de travail ..	Mémoire rémanente
1		1 %	0 %	0 %	0 %
2					
3	Totale:	4 Mo	460800 octets	1572864 octets	90784 octets
4	Occupée(s):	43471 octets	878 octets	730 octets	42 octets
5	Détails				
6	OB	>9852 octets	>266 octets		
7	FC	>20955 octets	>612 octets		
8	FC_Signalisation [FC14]	?	-		
9	FC_En_service [FC15]	4282 octets	138 octets		
10	FC_Convoyeur [FC16]	6822 octets	199 octets		
11	FC_Poids_LOG [FC36]	9851 octets	275 octets		
12	FB	?	?		
13	DB	6936 octets		730 octets	10 octets
14	Objets pour Motion Technology	-		-	0 octets
15	Types de données	902 octets			
16	Variables API	4826 octets			32 octets

Les Ressources sont accessibles via le menu: « Accessoires > Ressources » et affichent quel espace mémoire (quantitatif) est utilisé par un objet dans la CPU.

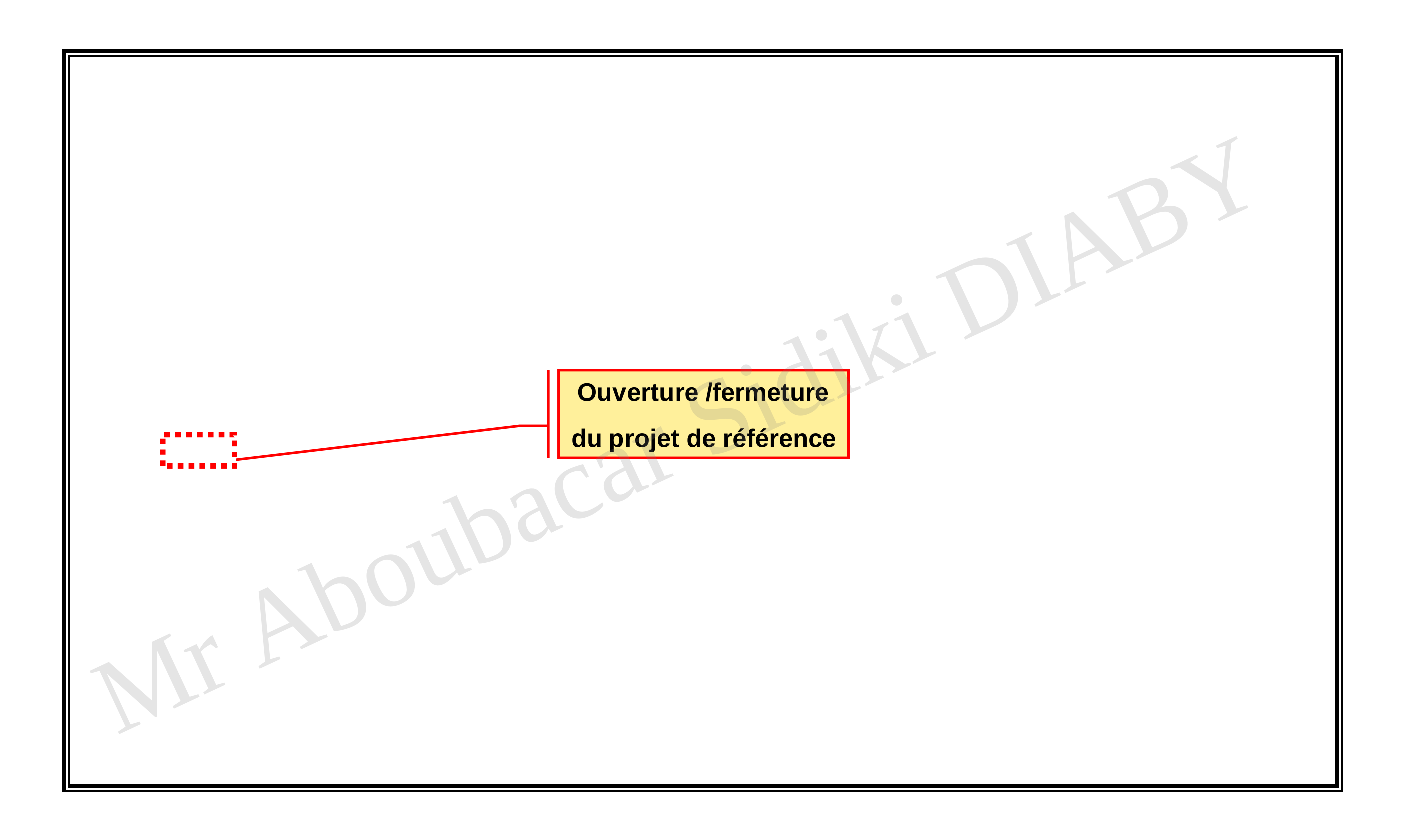
Remarque

Affichage de la mémoire de chargement utilisé dans la CPU

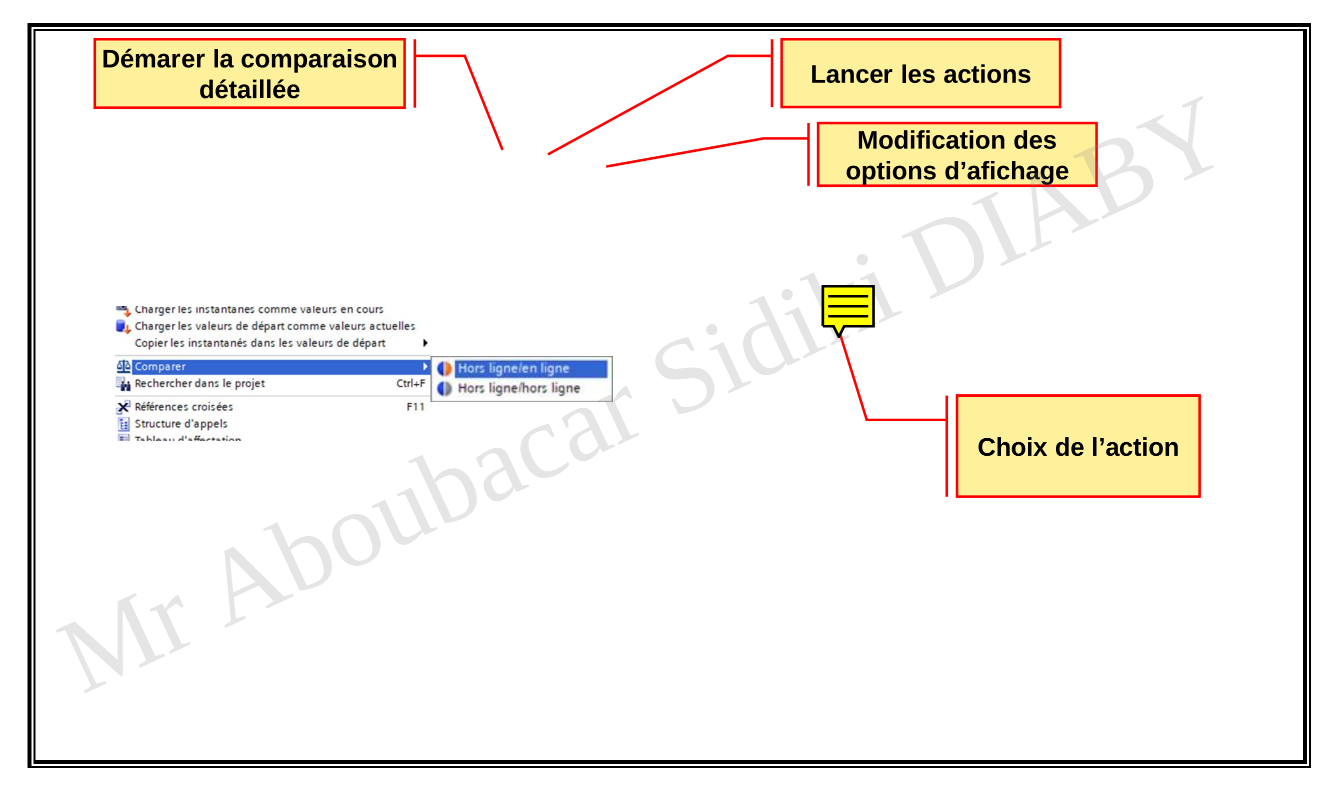
Veuillez noter que la somme de la mémoire de chargement utilisée ne peut pas être déterminée exactement si l'ensemble des blocs n'a pas encore été compilé.

Dans ce cas un « > » placé devant la somme indique que la valeur pour la mémoire utilisée pourra être plus importante qu'indiquée car les blocs qui ne sont pas compilés ne sont pas comptabilisés.

15.11.7. Projet de référence



15.11.8. Comparer (1) Hors ligne / en ligne



Mode de comparaison

On dispose en principe de deux modes de comparaisons différents:

Comparaison Hors ligne/en ligne:

- Les objets du projet sont comparés aux objets de l'appareil correspondant. Une liaison en ligne avec l'appareil est nécessaire à cet effet.

Comparaison Hors ligne/hors ligne:

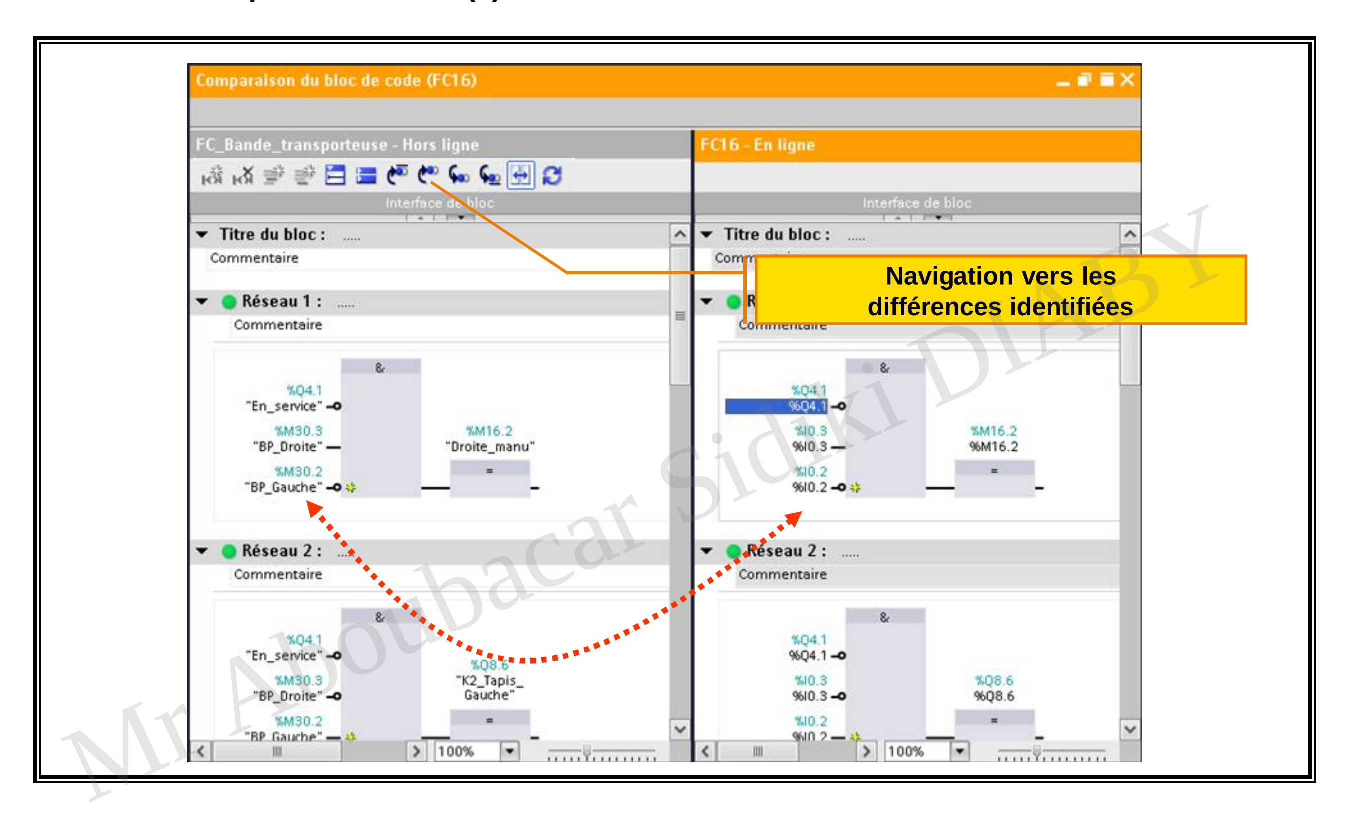
- Les objets de deux appareils d'un même projet ou de projets différents sont comparés.

Symbole de l'affichage du résultat

Le résultat de la comparaison est représenté à l'aide de symboles.
Le tableau suivant montre les symboles des résultats d'une comparaison Hors ligne/en ligne:

Symbole	Signification
	Le dossier contient des objets dont les versions hors ligne et en ligne diffèrent
	Résultat de la comparaison inconnu
	Les versions en ligne et hors ligne de l'objet sont identiques
	Les versions hors ligne et en ligne de l'objet diffèrent
	Objet uniquement présent hors ligne
	Objet uniquement présent en ligne

15.11.8.1.Comparer des blocs(2) Vue détaillée



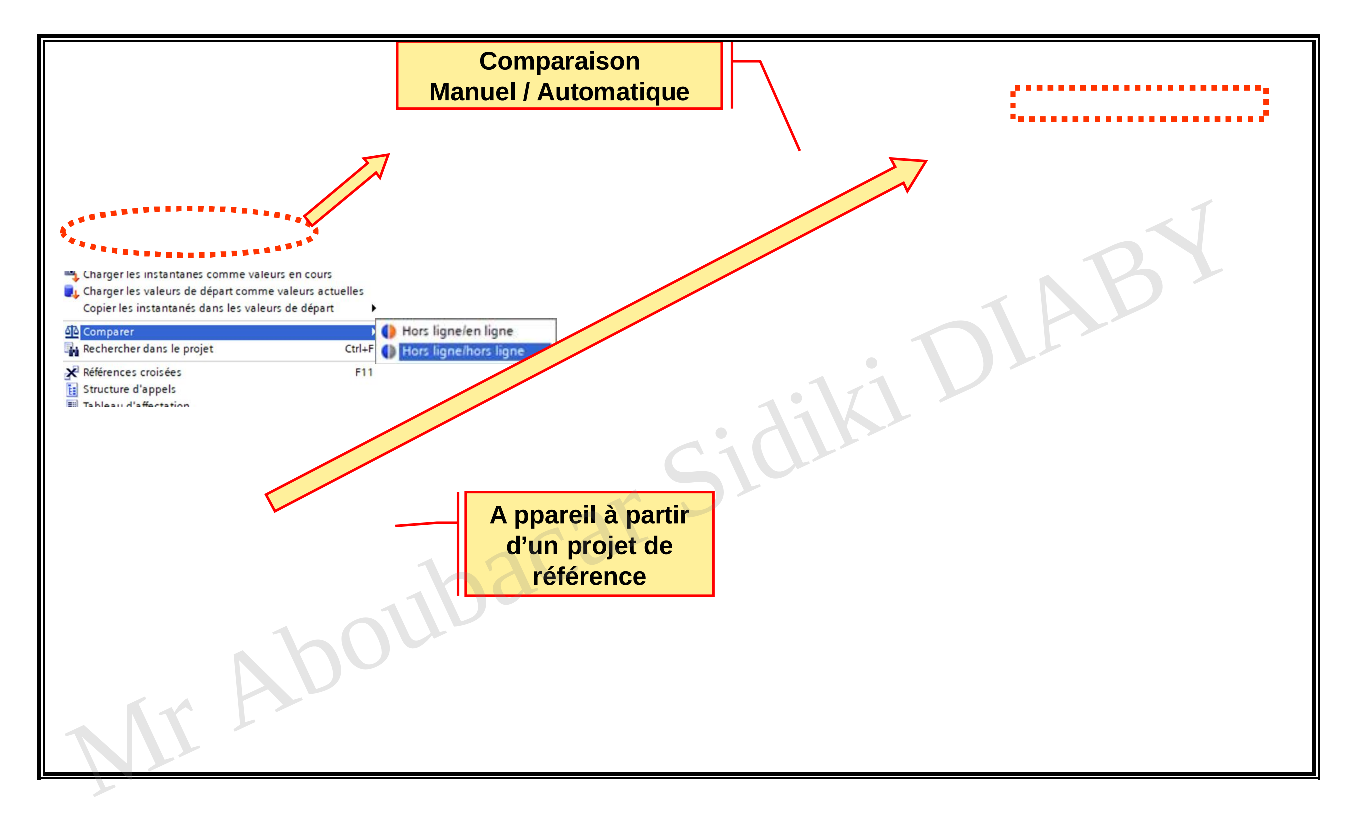
Comparaison détaillée

La comparaison détaillée permet d'identifier avec précision les endroits où les versions hors ligne et en ligne d'un bloc diffèrent. Les identifications suivantes sont utilisées:

- Les lignes contenant des différences sont affichées en gris.
- Les opérandes et les opérations différents sont affichés en vert.
- Si le nombre de réseaux est différent, des pseudo-réseaux sont insérés pour permettre une représentation synchronisée des réseaux identiques. Ces pseudo-réseaux sont affichés en gris et leur ligne de titre contient le texte « Aucun réseau correspondant n'a été trouvé ».les pseudo-réseaux ne peuvent pas être édités.
- Si l'ordre des réseaux est modifié, des pseudo-réseaux sont insérés aux endroits correspondants. Ces pseudo-réseaux sont affichés en gris et leur ligne de titre contient le texte « Les réseaux ne sont pas synchronisés ».Le pseudo-réseau contient en outre un lien « Aller au réseau <n°>, qui permet d'atteindre le réseau correspondant.

15.11.9. Comparer (3) Logiciel Hors ligne / En ligne





Comparaison logicielle Hors ligne / En ligne:

- Les objets de deux appareils d'un projet,
- Les blocs de deux appareils d'un projet,
- Les blocs d'un appareil,
- Les objets de différents projets,
- Les blocs de différents projets,

Sont comparés entre eux.

Pour cela vous pouvez choisir entre la comparaison



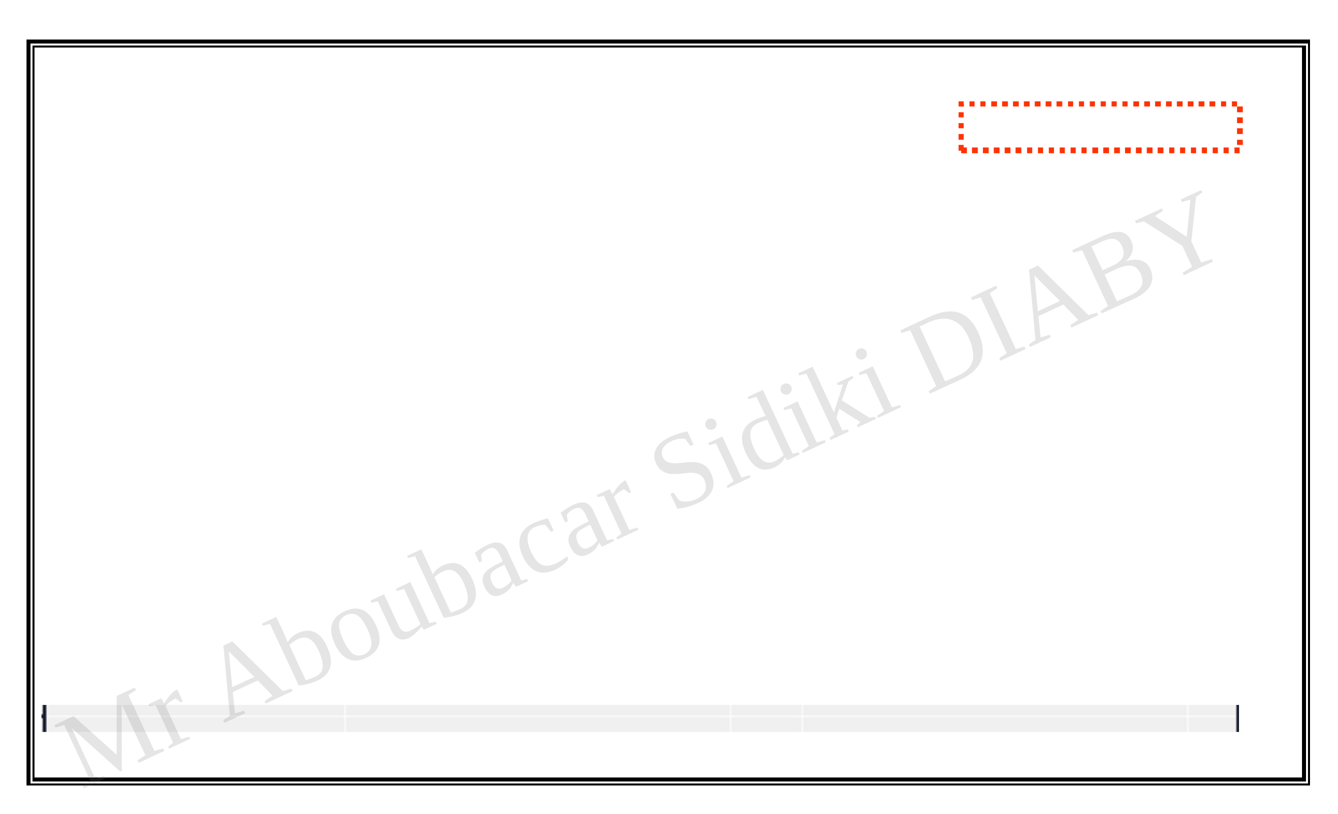
Comparaison automatique:

Les blocs et les objets de même type et de même nom sont comparés

Comparaison manuelle:

Vous pouvez choisir les blocs à comparer entre eux. De cette manière vous pouvez comparer tous les blocs.

Comparer (4) – Matériel Hors ligne / En ligne

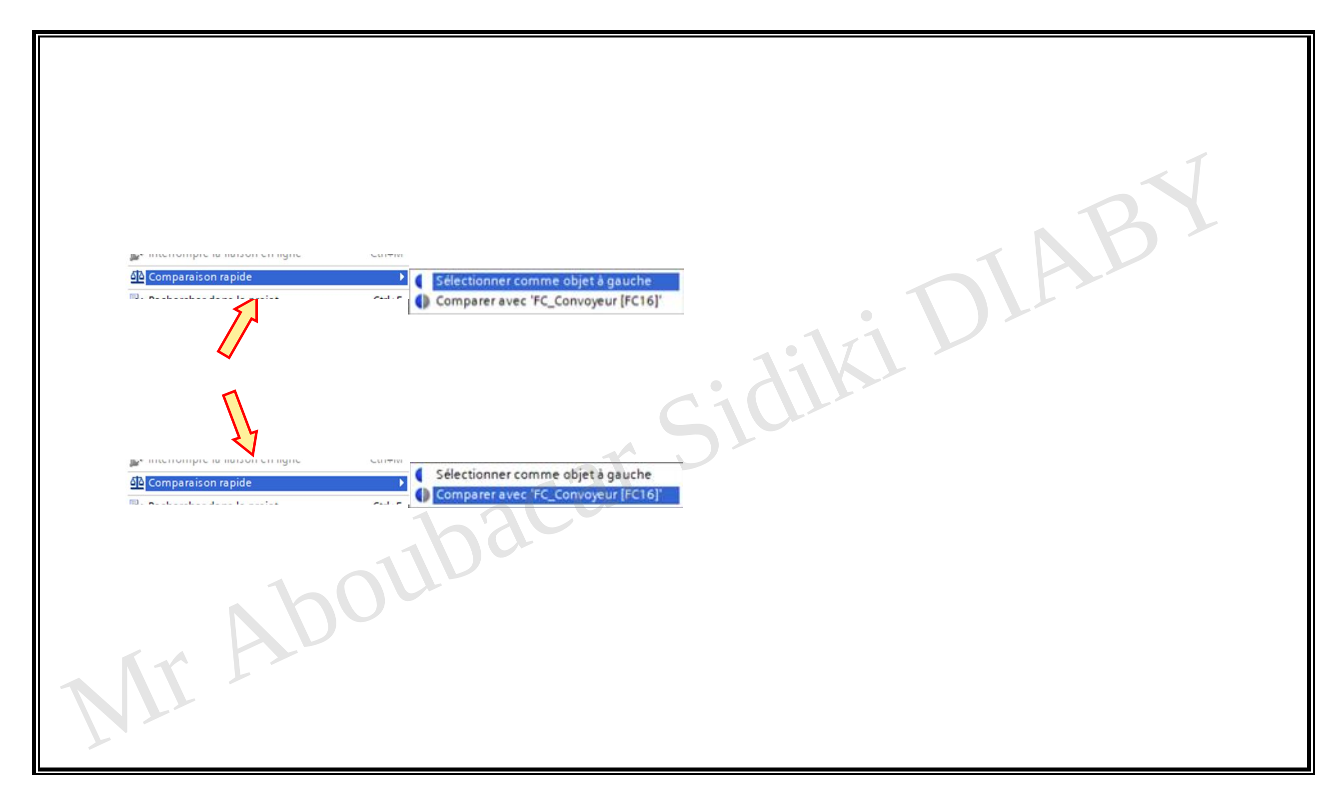


Comparaison du matériel Hors ligne / En ligne:

En complément, il est possible de comparer le matériel de deux appareils ou des modules d'un appareil.

Mr Aboubacar Sidiki DIABY

15.11.10. Comparer (5) Comparaison rapide de bloc



Hors ligne / hors ligne

Pour lancer une comparaison détaillée hors ligne / hors ligne d'un bloc directement dans le navigateur de projet, suivez ces étapes:

- Clic droit pour le bloc que vous voulez comparer. Le bloc pourra également appartenir à un projet de référence.
- Sélectionnez « Comparaison rapide » > « Sélectionner comme objet à gauche » dans le menu contextuel.
- Clic droit pour le bloc que vous voulez comparer avec le bloc sélectionné précédemment.
- Sélectionnez « Comparaison rapide » > Comparer avec « bloc sélectionné ». « bloc sélectionné » correspond à l'objet de la comparaison à gauche.

Hors ligne / en ligne

Pour lancer une comparaison En ligne / hors ligne pour un bloc directement dans le navigateur du projet, suives ces étapes:

- Etablissez une connexion En ligne avec l'appareil qui contient le bloc.
- Clic droit pour le bloc que vous voulez comparer avec son objet en ligne.
- Sélectionnez « Comparaison rapide » > Comparer avec l'objet en ligne.



15.11.11. Exercice 4: Test de la marche par impulsions du moteur



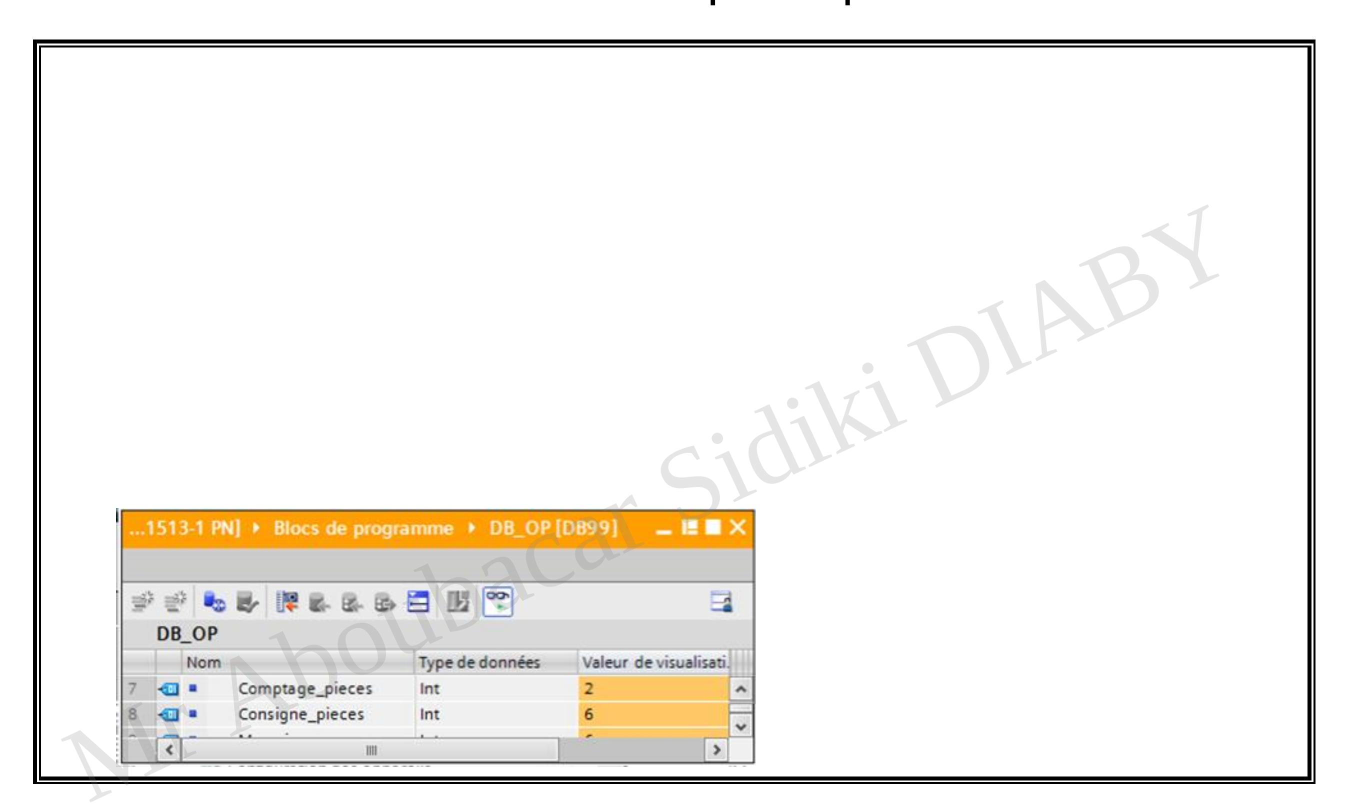
Enoncé

La fonction « Marche par impulsions du moteur » ne fonctionne pas. L'utilisation combinée des fonctions PG « Visualiser le bloc » et « Table de visualisation », montre qu'il doit y avoir une double affectation de la sortie « K_Droite » (Q4.5). L'exercice consiste à rechercher dans l'ensemble du programme utilisateur toutes les instructions qui accèdent en écriture à cette sortie.

Marche à suivre:

1. Redémarrez la CPU
2. Mettez l'installation à l'arrêt via le pupitre tactile
3. Ouvrez le bloc « FC_Convoyeur » (FC16) et activez la fonction de test « visualiser »
4. Dans le navigateur de projet, créez une nouvelle table de visualisation sous « Tables de visualisation et forçage » et observez la sortie « K_Droite » (Q4.5).
5. Affichez l'éditeur de bloc avec le « FC_Convoyeur » (FC16) et la table de visualisation l'un en dessous de l'autre en divisant la zone de travail (voir figure)
6. Interprétez les différents affichages d'état des deux fonctions de test
7. Localisez la double affectation sur la sortie « K_Droite » (Q4.5) à l'aide des données de référence et corrigez l'erreur.
8. Chargez les blocs modifiés dans la CPU et vérifiez la fonction.
9. Sauvegardez votre projet.

15.11.12. Exercice 6 : Vérification du compteur de pièces



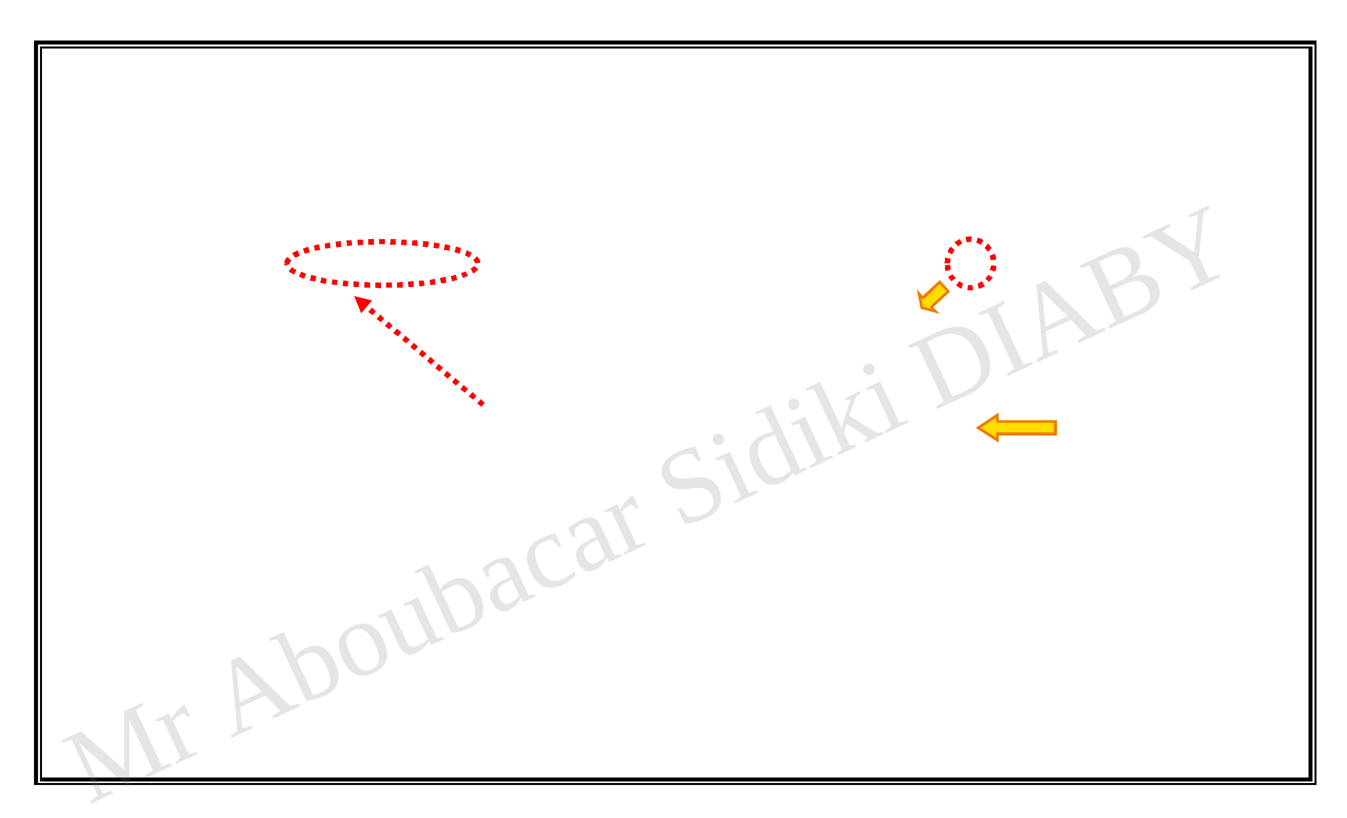
Enoncé

La recherche de la troisième erreur logicielle vous permettra d'utiliser la fonction de test "*Visualiser le bloc de données*" en simultané avec un bloc de code

Marche à suivre

1. Ouvrez le bloc « FC_Compteur_Add » FC19 puis mettez en visualisation dynamique
2. Ouvrez le bloc « DB_OP » DB99 puis mettez en visualisation dynamique
3. Sortez le de la fenêtre de travail en cliquant sur

15.11.13. Exercice 7 : Tester la surveillance du défaut 3



Enoncé

La recherche de la troisieme erreur logique vous permettra de paramétrer un déclenchement sur un appel déterminé du bloc comme le montre la figure ci-dessus.

Test fonctionnel 3

Vérifiez si *Défaut 3* (bouton I 0.6 du simulateur) est visualisé par le clignotement de la LED Q 0.4 sur le simulateur et si après validation par le bouton I0.7 du simulateur, la LED passe en continu.

Marche à suivre

1. Ouvrez le bloc FB20
2. Mettez vous en visualisation dynamique sur ce bloc
3. Soit par l'icône ou via la TaskCard à Tester à Environnement d'appel, reglez l'environnement comme le montre la figure ci-dessus

15.12. Fonction d'analyseur: TRACE



Fonction d'analyseur « Trace »

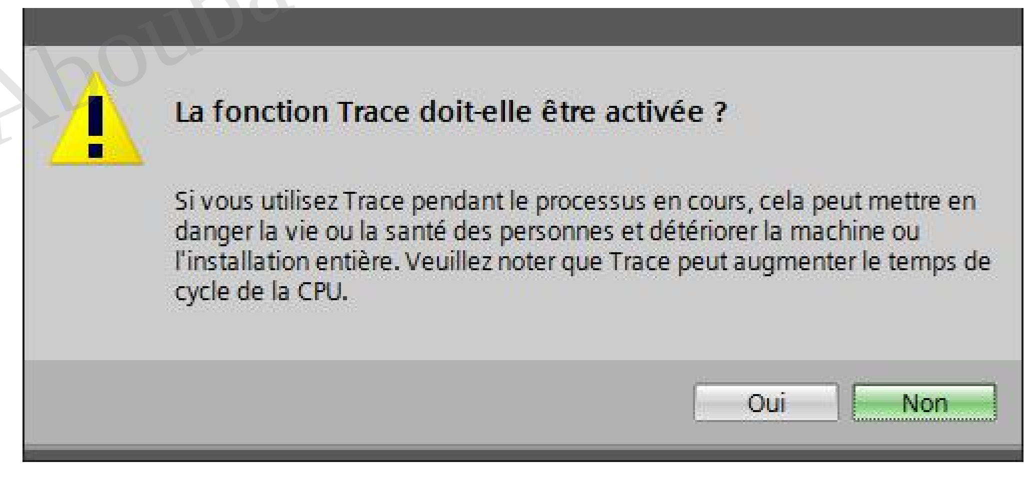
Dans une « TRACE » vous pouvez mémoriser l'évolution d'une ou de plusieurs variables de la CPU (max. 16). Dans STEP7, on peut représenter une « TRACE » sous forme graphique.

Le nombre de « TRACE » est fonction de la CPU en place.
Suivant la CPU on dispose des quantités de mémoires internes « TRACE » de 512 Ko suivantes.

- S7-1200 : 2x TRACE (FW ^{3} V4.0)
- S7-1517 4x TRACE, S7-1518 8x TRACE
- Par Trace il est possible d'enregistrer au maximum 16 signaux ou variables API.

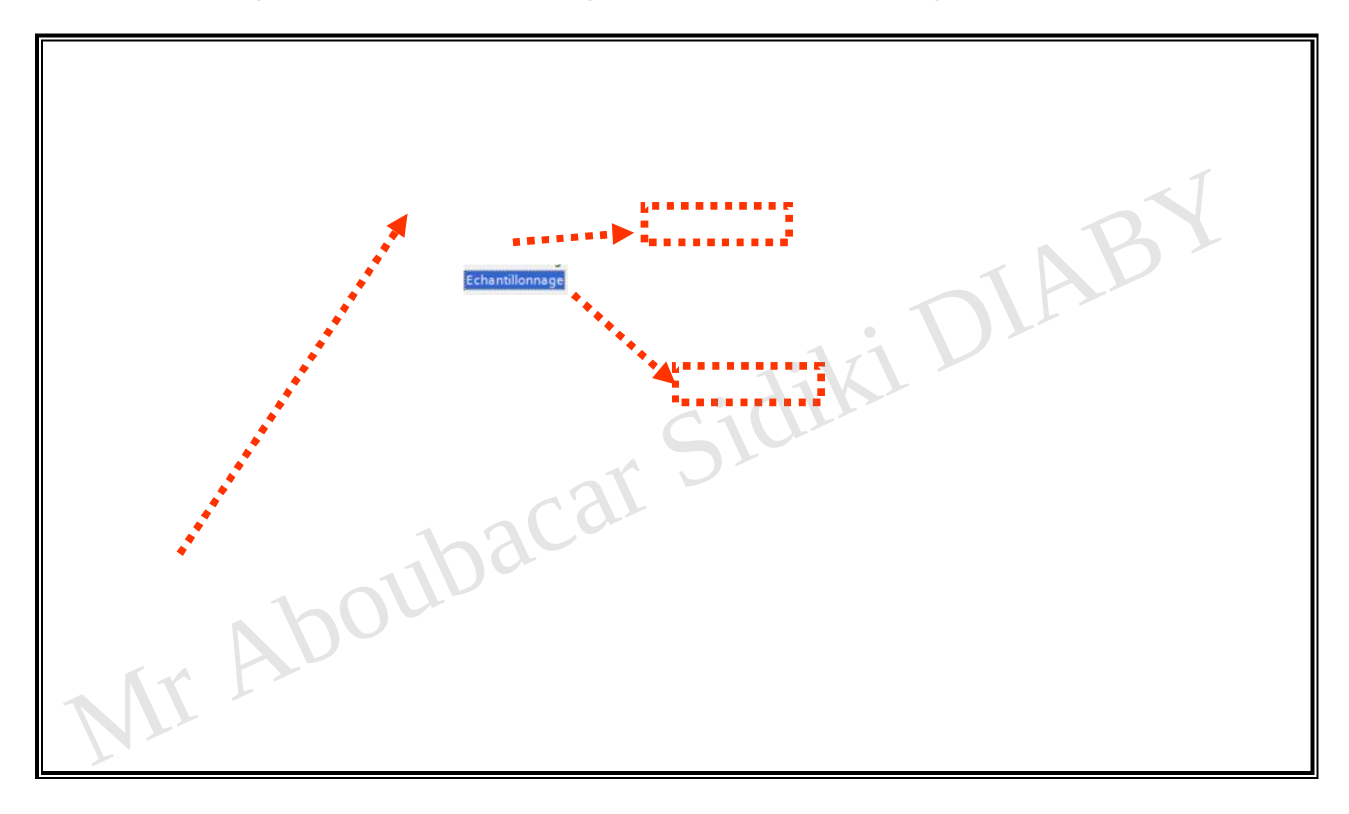


Attention!
Veuillez respecter l'avertissement ci-dessous avant d'activer « Trace » sur la CPU





15.12.1. Configurer une TRACE – Signaux et échantillonnage



Trace: Signaux

Les variables API globales d'un type de données élémentaires peuvent être enregistrées.

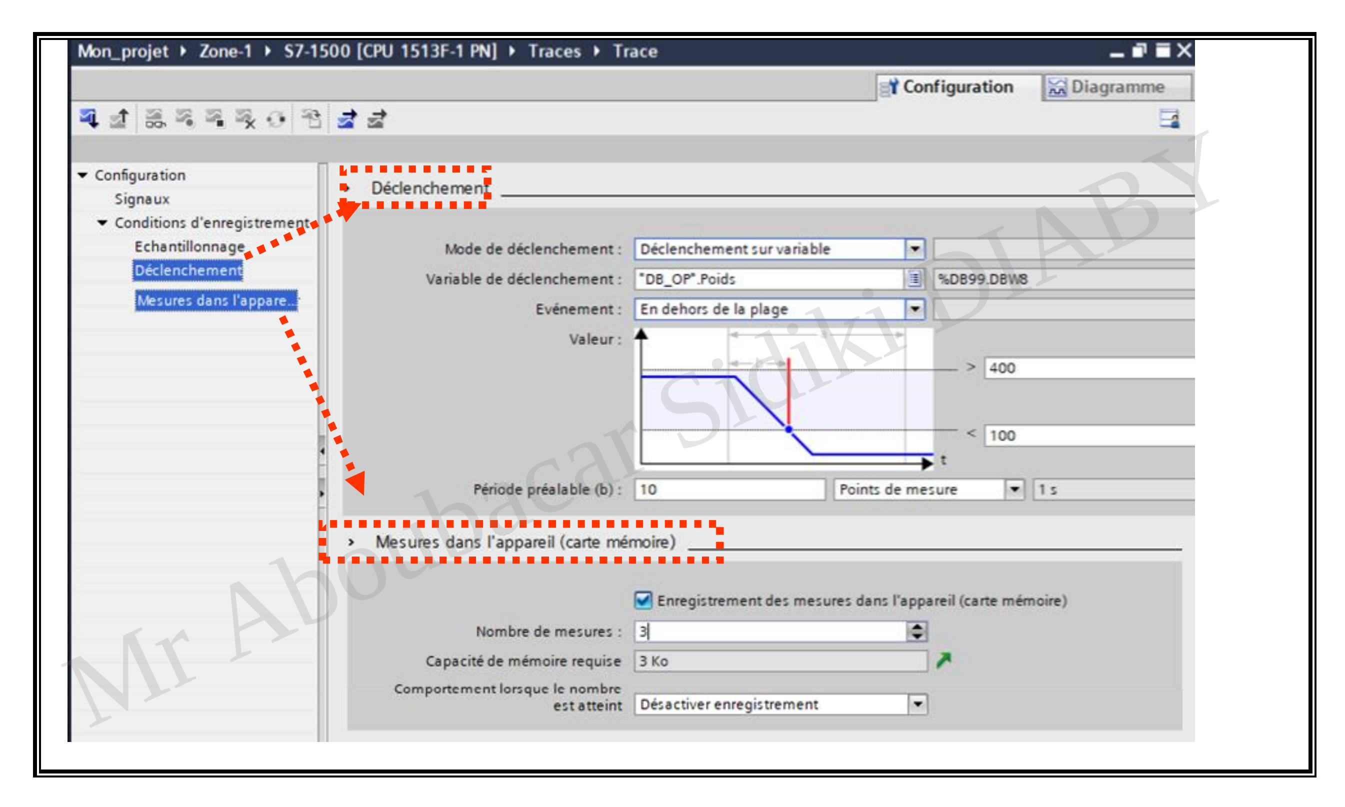
Trace : Echantillonnage et Durée d'enregistrement

A ce niveau vous définissez la fréquence ou l'intervalle de temps séparant l'enregistrement des variables. A partir de ces intervalles d'échantillonnage et du type de données ou de la dimension de la variable, vous obtenez la durée maximale de l'enregistrement de la Trace car la place mémoire attribuée pour l'enregistrement est limitée.

- Taille maximale de l'occupation mémoire d'une Trace: 512 K octets – 30 octets (gestion interne) = 524258 octets
  Chaque point de mesure est enregistré avec l'horodatage (8 octets). Le résultat …
- Nombre d'octets par point de mesure = 8 octets + nombre d'octets par valeur de mesure (pour un booléen, le nombre d'octets par valeur mesurée est de 1 octet)
- Exemple: Trace avec 1 variable INT et un intervalle d'échantillonnage de 100ms
  - Trace du type de données INT -> 8+2 octets/point de mesure -> 52425 points de mesure possibles
  - Intervalle d'échantillonnage = 100ms -> 10 valeurs de mesures par seconde
  - -> durée maximale de l'enregistrement: 52425 points / 10 points par seconde = 5242 secondes

Mr Aboubacar Sidiki DIABY

15.12.2. Configurer TRACE – Déclenchement et Enregistrement des mesures dans l'appareil



Déclenchement de l'enregistrement de Trace

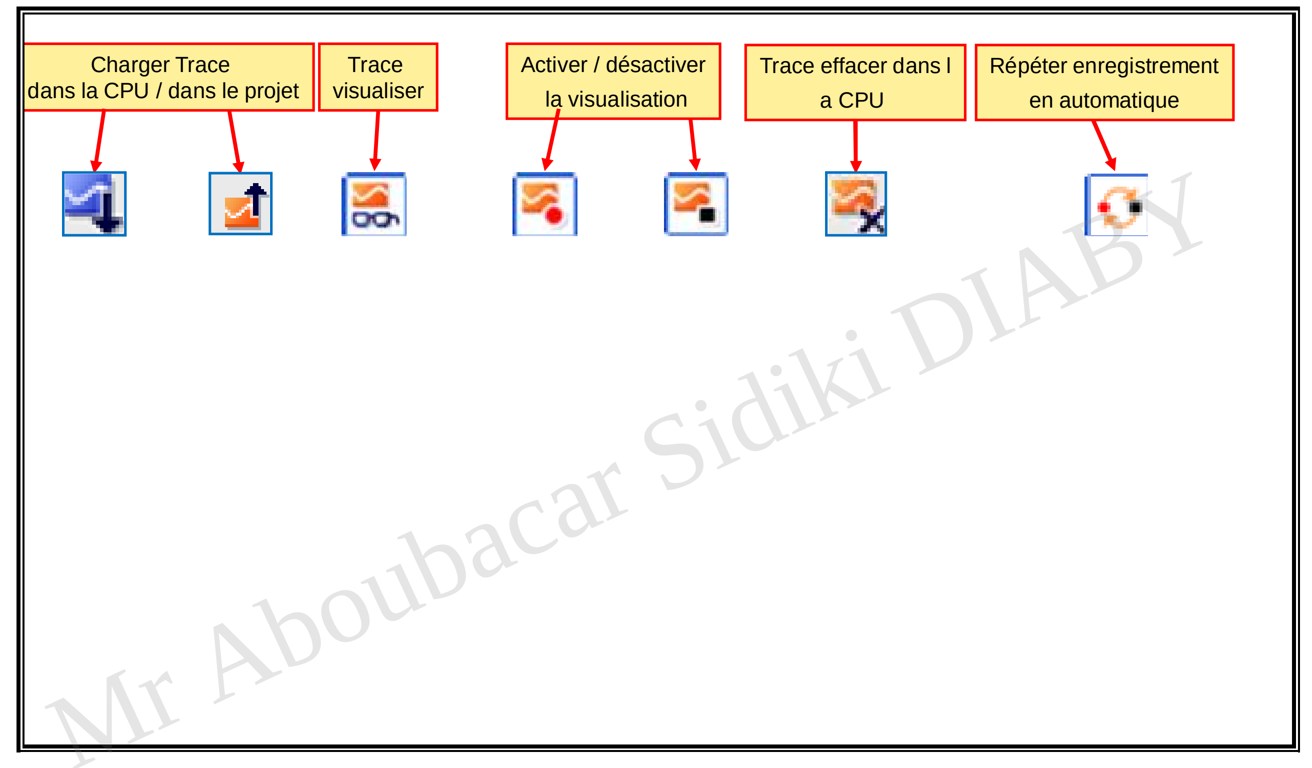
Le déclenchement de l'enregistrement d'une Trace démarre la mesure et l'affichage des signaux de la Trace, cependant, il ne s'agit pas de sauvegarde définitive des valeurs car elles sont enregistrées dans une mémoire tampon circulaire et constamment écrasées par les nouvelles valeurs. Ce n'est que lorsque les conditions de déclenchement sont remplies que les valeurs temporaires sont enregistrées définitivement et ainsi elles ne sont plus écrasées aves de nouvelles valeurs. L'évènement de déclenchement est dépendant du type de données de la variable de déclenchement. L'enregistrement de la Trace s'arrête dès que la durée maximale d'enregistrement configurée dans l'échantillonnage est atteinte.

En définissant une période préalable au déclenchement vous déterminez la quantité de points de mesure à enregistrer définitivement avant le déclenchement.

Mesures dans l'appareil (Carte Mémoire)

Les mesures effectuées peuvent être enregistrés sur la carte mémoire afin de démarrer une nouvelle opération. Dans « enregistrement des mesures dans l'appareil » vous définissez le nombre de mesures à réaliser. En complément vous définissez si l'enregistrement le plus ancien doit être écrasé lorsque le nombre de mesures est atteint ou s'il faut arrêter l'enregistrement.

15.12.3. TRACE: charger dans la CPU, l'activer, visualiser





Charger une Trace dans la CPU / dans le projet

Après avoir configuré la Trace hors ligne ou dans le projet, il faut la charger dans la CPU, car ce n'est pas l'outil d'ingénierie qui exécute la fonction mais la CPU.

Seules les Traces présentes en ligne dans la CPU peuvent être chargées de la CPU vers le projet.

Visualisation d'une Trace

Visualise l'état d'une Trace de la CPU:

- Inactif (la Trace est chargée dans la CPU, mais n'a pas encore été activée)
- En attente de déclenchement (la Trace est activée dans la CPU, mais la condition de déclenchement n'est pas encore réalisée)
- Enregistrement en cours (la Trace est activée dans la CPU et l'enregistrement est en cours)
- Enregistrement terminé (la Trace est activée dans la CPU mais l'enregistrement est déjà terminé

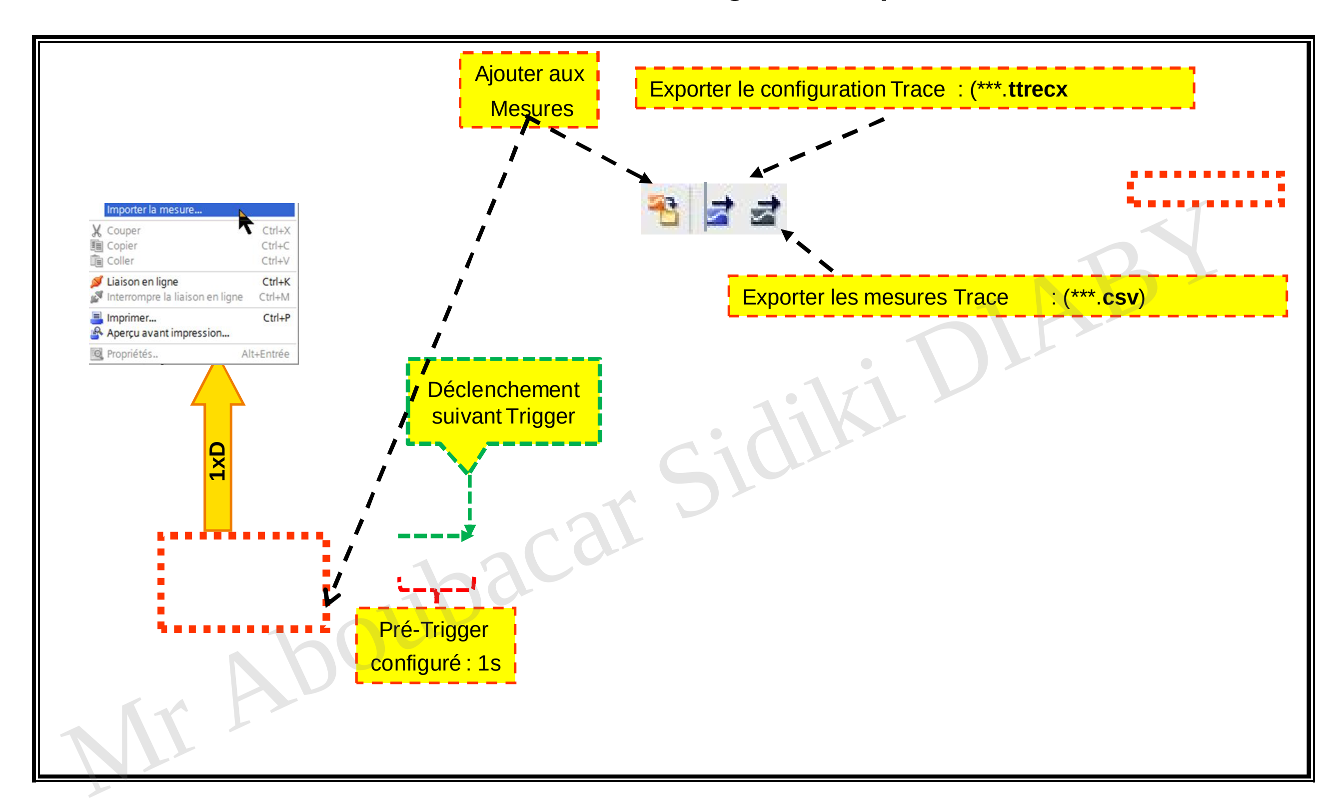
Activation de l'enregistrement  **/ désactivation** 

La fonction « activation de l'enregistrement » active la Trace, c'est-à-dire que la mesure et l'enregistrement des signaux TRACE sont démarrés immédiatement, même si la condition de déclenchement éventuelle n'est pas encore remplie. Les valeurs enregistrées sont visualisées en continu et déposées dans une mémoire tampon circulaire, écrasée en permanence par des nouvelles valeurs. Lorsque les conditions de déclenchement sont vérifiées alors les valeurs ne sont plus écrasées et restent mémorisées. A partir de cet instant l'enregistrement est prolongé jusqu'à l'écoulement de la durée maximale d'enregistrement.

Une nouvelle activation de Trace écrase l'enregistrement en ligne déjà présent dans la CPU.

La fonction « Désactivation de l'enregistrement »permet de désactiver une Trace en « Attente de déclenchement » ou d'interrompre un enregistrement déjà en cours.

15.12.4. TRACE dans STEP7: évaluer, sauvegarder, exporter



Visualisation et exploitation de TRACE : Diagramme

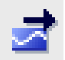
L’éditeur de Trace permet, en étant « En ligne », de visualiser l’enregistrement de la Trace mémorisé actuellement dans la CPU au niveau de la fenêtre « Diagramme ».

Des enregistrements Trace sauvegardées « Hors ligne » dans le projet peuvent être visualisés en double-cliquant sur « Mesures ».

Enregistrer une Trace dans le projet 

Cette fonction permet de transférer les Traces sauvegardées en ligne dans la CPU vers le projet hors ligne. Une trace peut uniquement être mémorisée dans un projet si elle est pleine ou si l’enregistrement a été arrêté.

Exporter la configuration et l’enregistrement d’une Trace

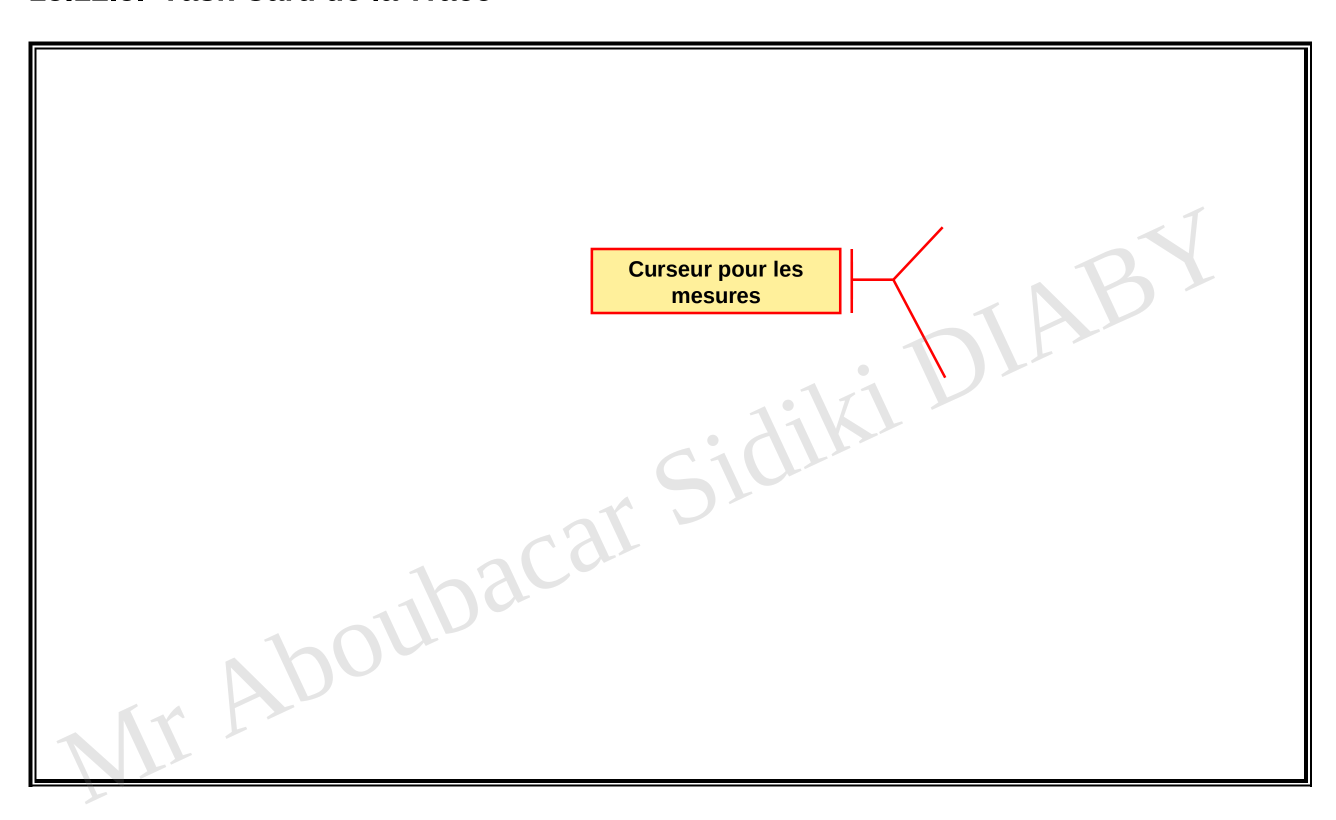


- **La Trace est ouverte dans l’éditeur de Trace (dans la figure « Poids »):**
  La configuration de la Trace est exportée sous forme de fichier TTCRX
  Cette fonction exporte la configuration d’une Trace, qui peut par la suite être importée dans d’autres projets.



- **L’enregistrement d’une trace est ouvert dans l’éditeur de Trace (dans la figure par exemple « Poids_2 »): Un enregistrement de Trace est exporté sous forme de fichier CSV**
  Cette fonction exporte un enregistrement Trace qui peut, par la suite, être exploité sous MS Excel par exemple.

15.12.5. Task Card de la Trace



Palette « Curseur de mesure »

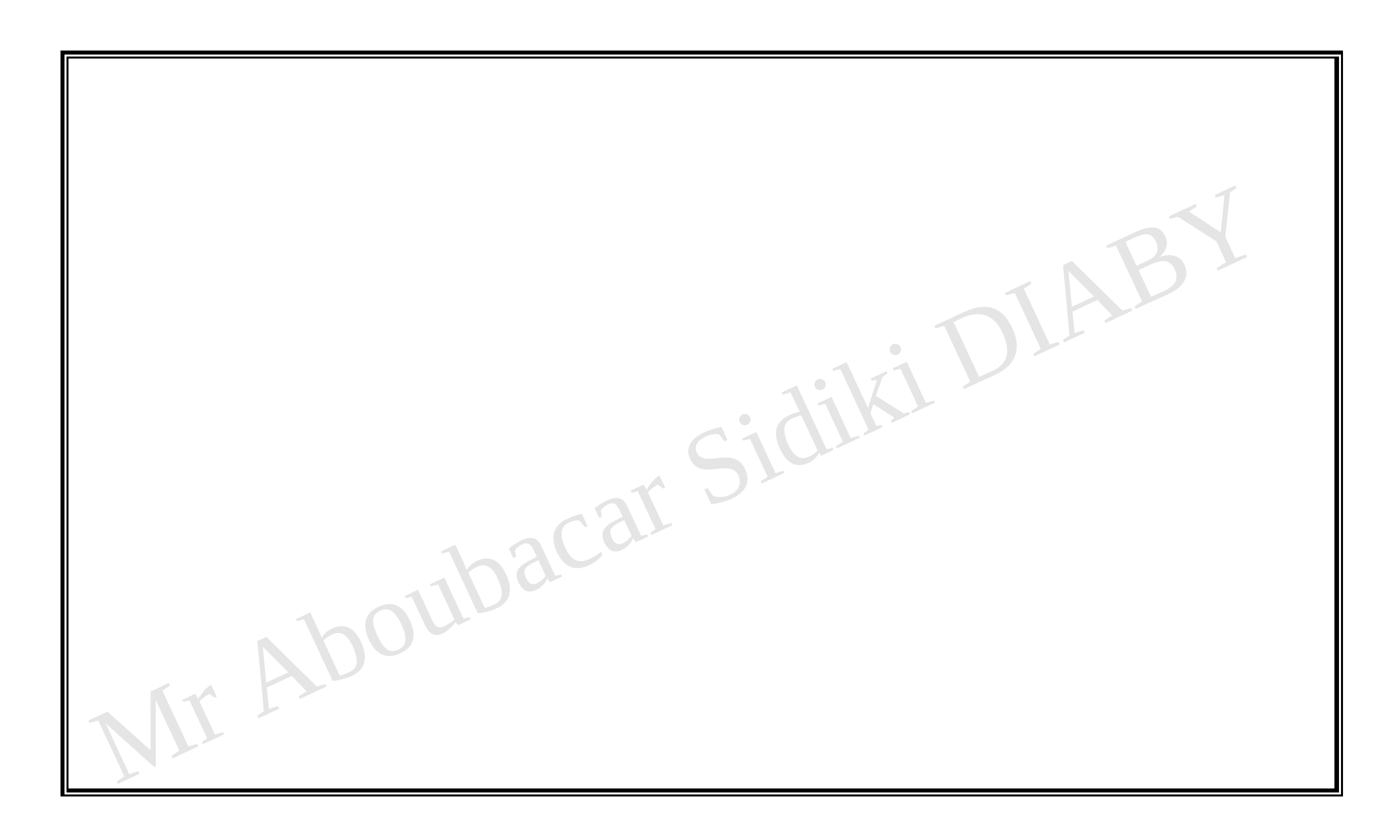
La palette « Curseur de mesure » affiche la position des curseurs de mesure dans la graphique ainsi que les valeurs aux intersections.

Palette « Instantanés »

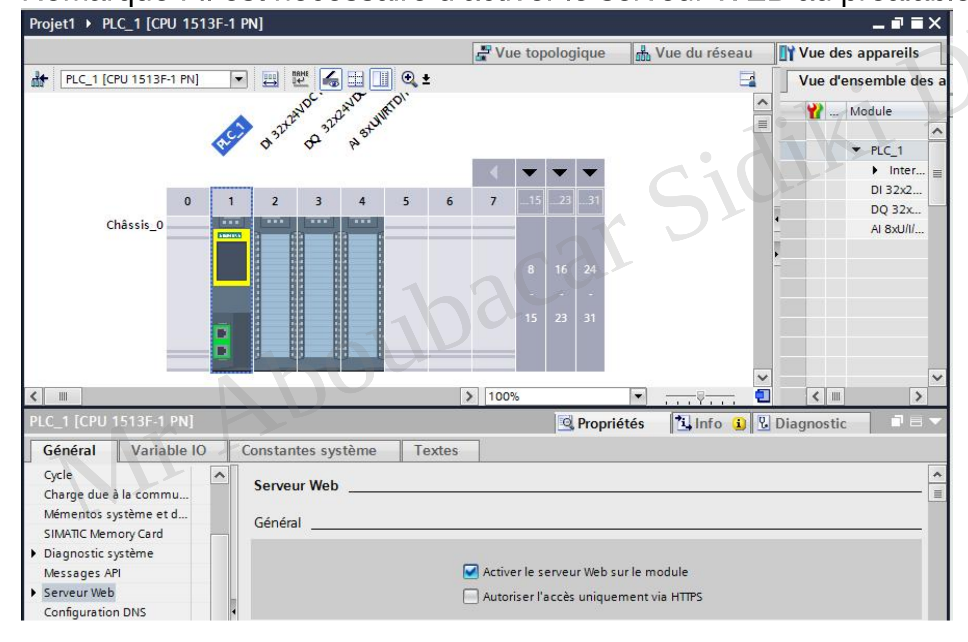
La palette « instantanés » permet la sauvegarde et la récupération de différentes vues d'une mesure.

Un instantané est créé à partir de la vue courante sous « Diagramme ». Les instantanés sont sauvegardés dans les mesures du projet.

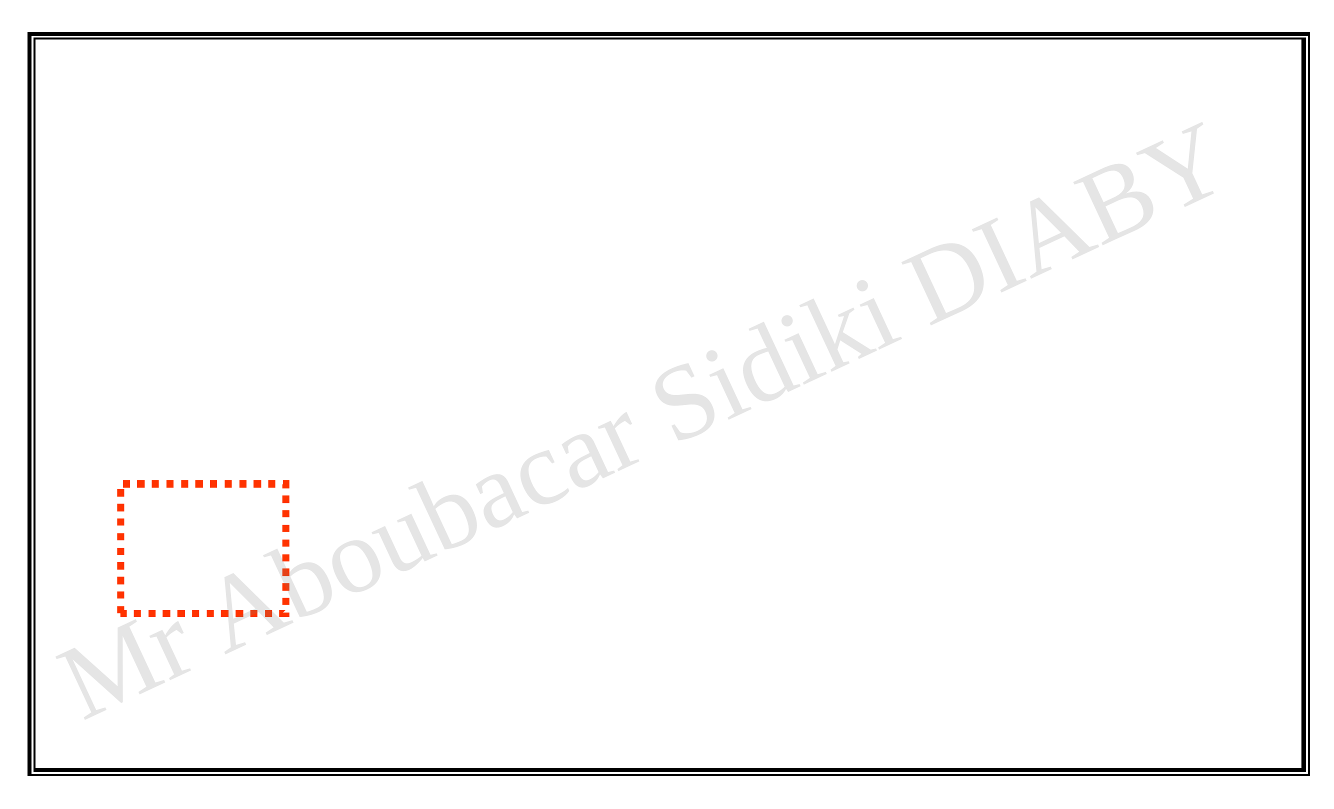
15.12.6. TRACE : visualisation de la trace dans le serveur WEB



Il est possible d'exploiter les traces directement dans le serveur WEB sous condition d'avoir une CPU avec un firmware v2.0 minimum et de l'avoir configurée avec TIA Portal v14.
Remarque : Il est nécessaire d'activer le serveur WEB au préalable :



15.12.7. Exercice 7: créer, visualiser et enregistrer une TRACE



Enoncé

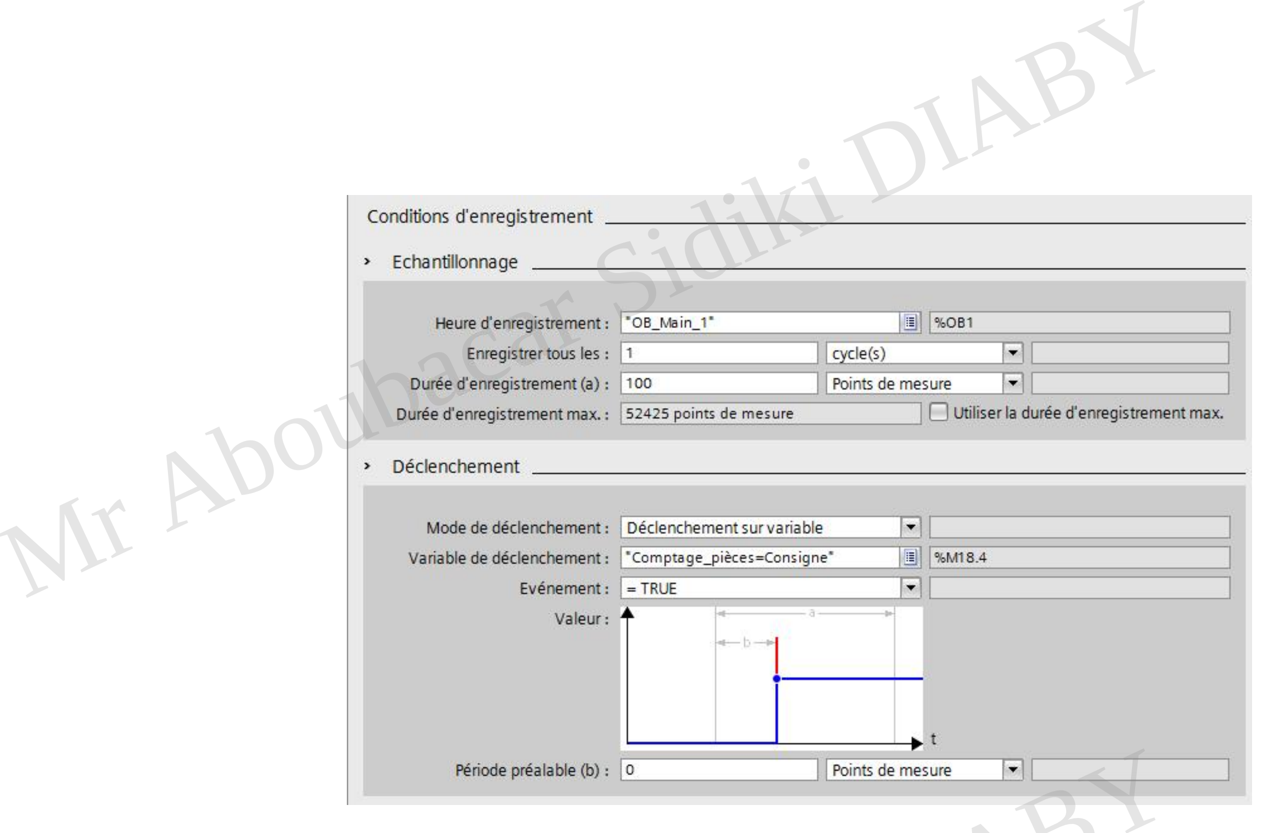
Réalisez une configuration d'une Trace avec les propriétés suivantes:

- Le signal Trace doit être la variable « Curseur_1 » (IW10)
- La variable doit être scrutée à chaque appel d'OB1.
- Les enregistrements des valeurs de la variable « Curseur_1 » doivent être démarrés dès que le compteur atteint sa valeur maximum
- La Trace doit être enregistrée pendant 100 cycles.

Suite page suivante

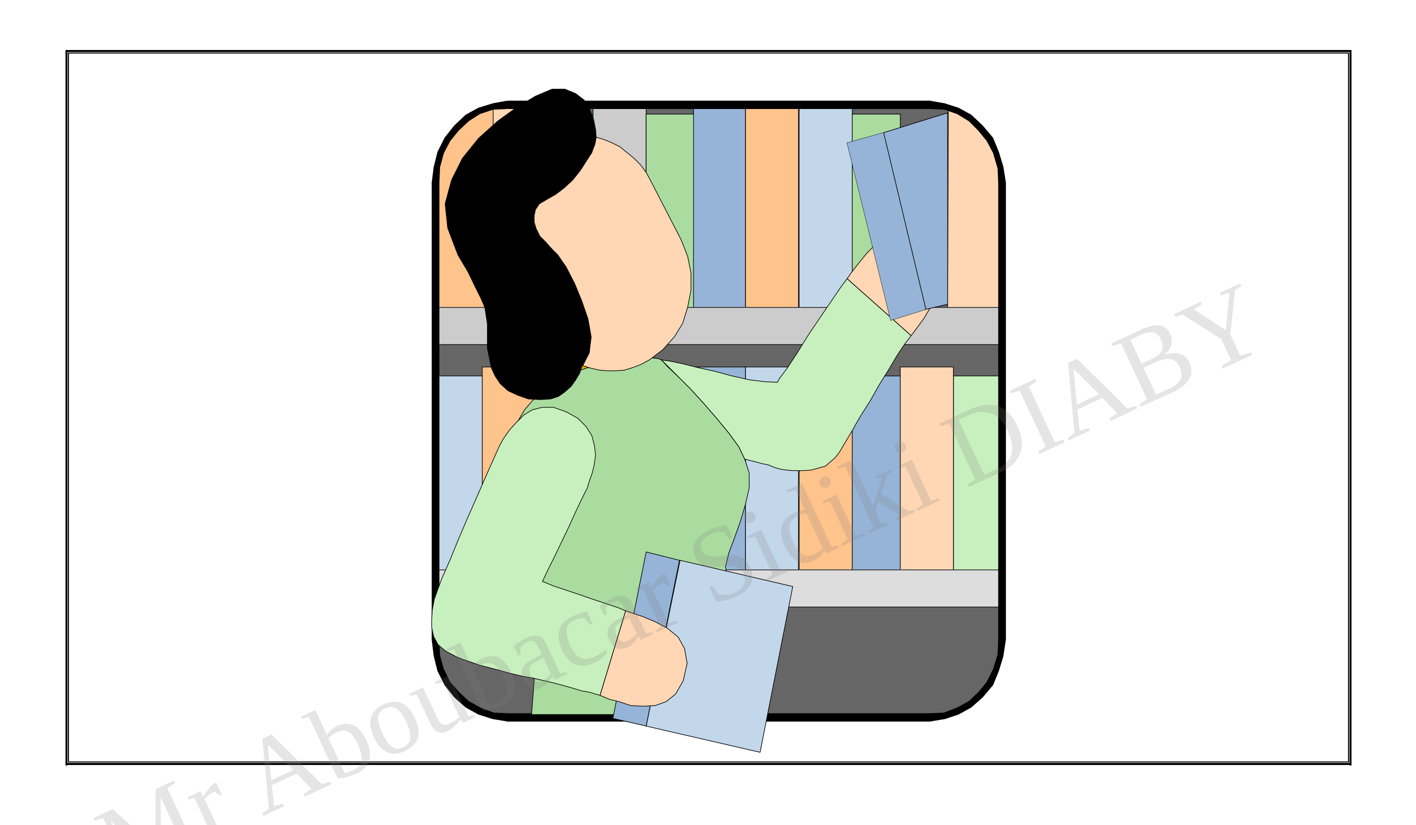
Marche à suivre

1. Ajoutez une Trace « Curseur » et confirez la comme indiqué ci dessous.

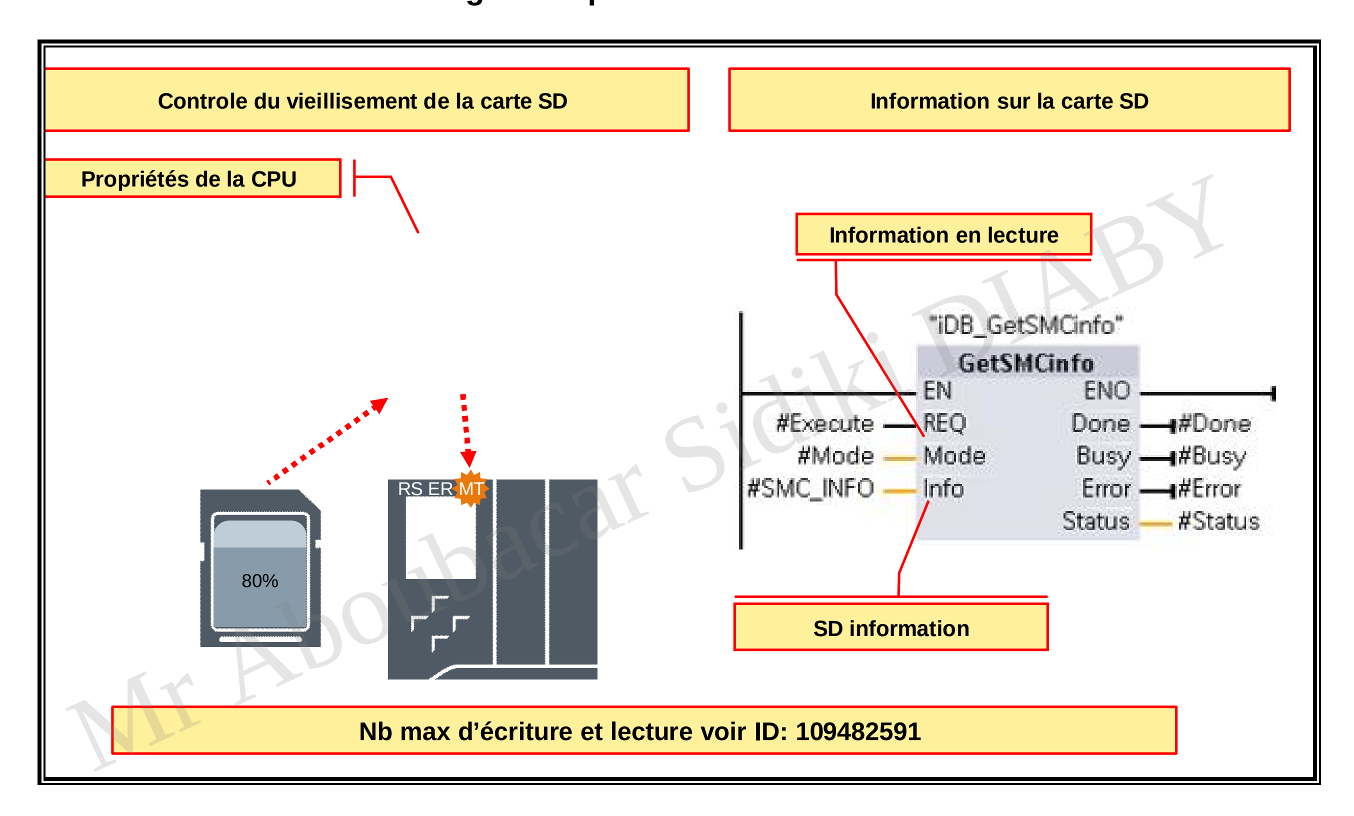


2. Chargez la Trace dans la CPU et activez la.
3. Réglez un poids non valable au niveau du curseur gauche du tableau de commande (<100 ou >400) et vérifiez si l'enregistrement de la Trace prend fin au bout de 10 secondes.
4. Sauvegardez la Trace enregistrée dans le projet sous « Curseur_1 » et démarrez un nouvel enregistrement.
5. Ajoutez encore deux autres enregistrements Trace : « Curseur_2 » et « Curseur_3 » pour les ouvrir ultérieurement et à partir du projet les exporter sous forme de fichier CSV.

Informations complémentaires



15.12.8. Informations de diagnostic pour Carte Mémoire SIMATIC



Vérification de la durée de vie

Génération d'un enregistrement de diagnostic lorsque la durée de vie retenue est dépassée (en pourcentage des cycles d'écriture/lecture garantis)

Information visuelle par la LED Maintenance

Instruction « GetSMCinfo »

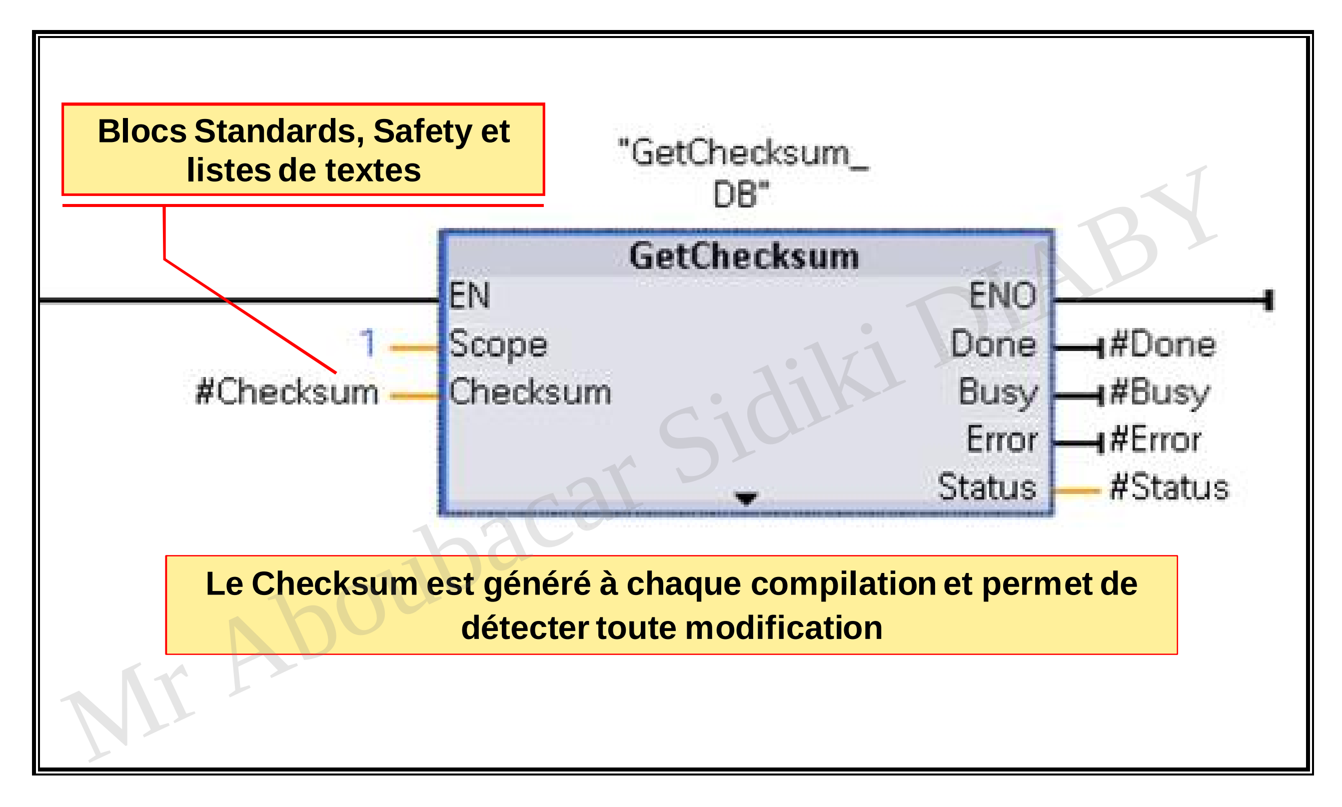
Avec l'instruction « GetSMCinfo » vous récupérez des informations au sujet de la carte mémoire SIMATIC insérée. L'information à extraire est sélectionnée par le paramètre « Mode ».

- 0: Taille de la mémoire en K Octets = 1024 octets)
- 1: Espace mémoire occupé en K Octets
- 2: Information pour la maintenance: Pourcentage de la durée de vie déjà utilisé en %
- 3: Valeur en pourcentage de la durée de vie pour laquelle la CPU créé une entrée dans le tampon de diagnostic, allume la LED maintenance et génère une alarme de diagnostic.

Remarque:

Voir **FAQ: 109482591** – « Comment calculer la durée de vie d'une carte mémoire du S7-1500/S7-1200 et ET200 ? »

15.12.9. Lecture du total de contrôle (Checksum)



Générer le total de contrôle (Checksum)

Les programmes API se voient attribuer automatiquement une somme de contrôle unique. Si le programme a été modifié il reçoit une nouvelle somme de contrôle. Si le programme API n'a pas été modifié et s'il est à nouveau recompilé, la somme de contrôle reste identique.

Si des modifications sont réalisées puis annulées par la suite, la somme de contrôle reste la même.

Charger le total de contrôle

La somme de contrôle est chargée dans la CPU avec le programme API, elle est disponible dans le programme En ligne. Les blocs modifiés ou générés pendant le traitement du programme (par exemple : « WRIT_DBL », « CREAT_DB » et « DELETE_DB ») ne modifient pas le somme de contrôle. Lors du chargement à partir de la CPU, la somme de contrôle n'est pas reprise dans le projet hors ligne car elle sera automatiquement régénérée lors de la prochaine compilation.

Evaluation du total de contrôle

La somme de contrôle est affichée dans les propriétés de la CPU (Propriétés > Général > Totaux de contrôle). A partir de là, vous pouvez les reprendre manuellement dans vos documents. Pour extraire le total de contrôle du programme vous disposez de l'instruction avancée « GetChecksum ».

Sommaire 16

16. Mise en service et intégration d’un variateur avec Startdrive

A l’issue du chapitre, vous allez...



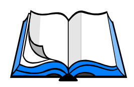
... savoir paramétrer et tester un variateur avec Startdrive

... savoir intégrer un variateur dans la configuration matérielle

… savoir restaurer les réglages usine sur un variateur

... savoir définir les paramètres de base via Startdrive

… savoir surveiller et commander un variateur

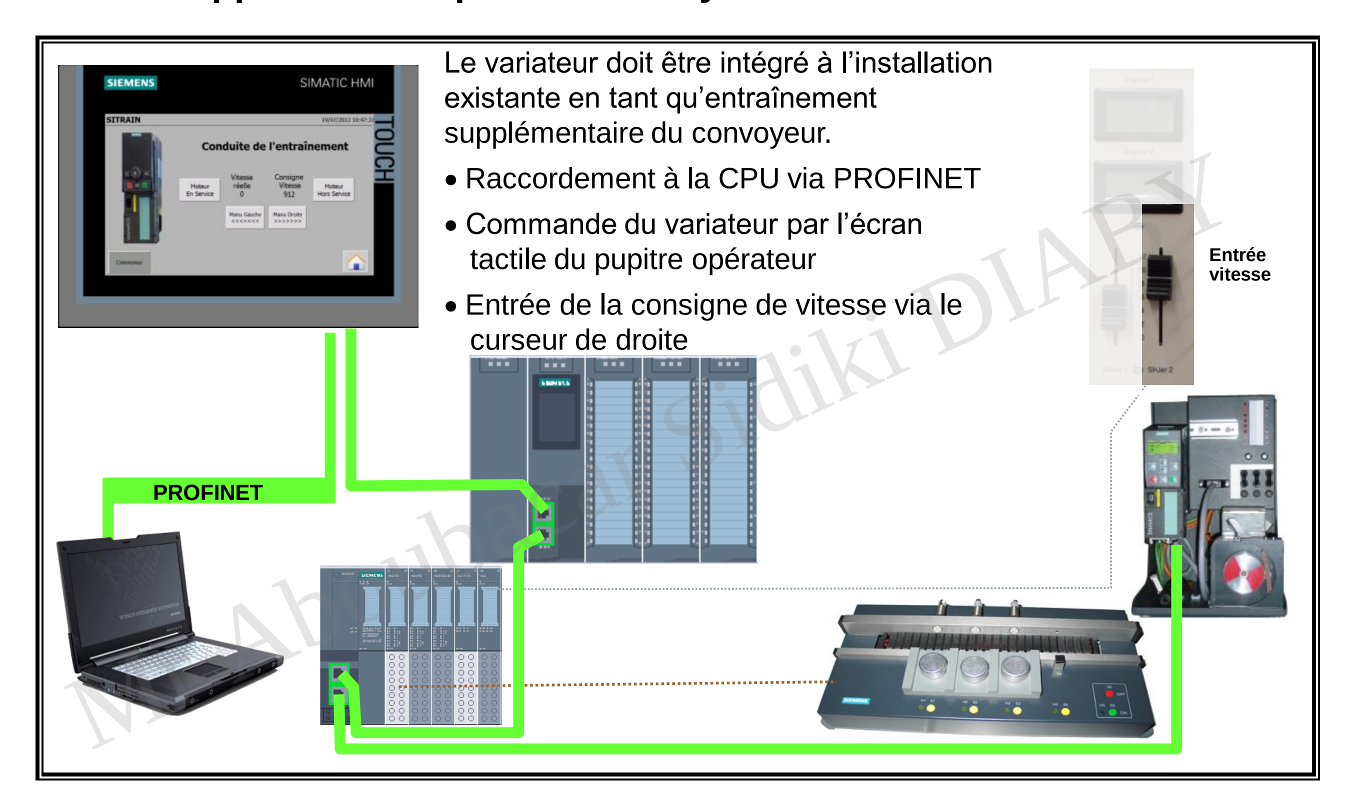


Ce chapitre présente l’utilisation d’un composant périphérique décentralisé.

A l’issue du chapitre, vous connaîtrez les principes d’un variateur de vitesse. Vous aurez une vue d’ensemble de la mise en service d’un système d’entraînement avec l’option d’ingénierie Startdrive.

Objectifs pédagogiques :
Savoir mettre en service un variateur de la gamme SINAMICS G120 et l’intégrer dans un programme S7.

16.1. Objectif du chapitre : utiliser un G120 comme entraînement supplémentaire pour le convoyeur



Enoncé

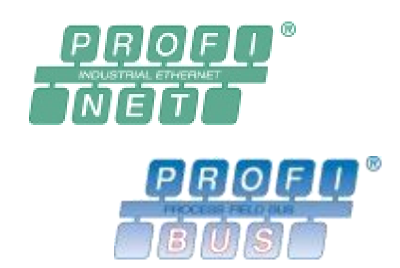
Imaginez que la variateur soit utilisé en tant qu'entraînement complémentaire de la maquette du convoyeur.

Le variateur doit être piloté comme suit:

- Quand « Q_En_Service » (Q0.1) est activé, le variateur G120 est aussi piloté en parallèle au moteur du convoyeur.
- La vitesse du G120 pourra être réglée avec le curseur à droite (potentiomètre) de la maquette.

16.2. Standard de communication PROFIdrive







- Communication industrielle homogène et continue
- Vaste domaine d'application
- PROFIdrive est le profil standard destiné aux entraînements utilisés en liaison avec les systèmes de communication PROFIBUS et PROFINET
- Méthode éprouvée pour le couplage simple et continu de systèmes d'entraînement et d'automates de différents constructeurs

La communication industrielle doit également satisfaire aux exigences d'interopérabilité. PROFIBUS et PROFINET constituent à ce titre des solutions optimales, car ils autorisent une parfaite opérabilité et une adaptation flexible aux applications. Ils s'imposent par ailleurs face aux autres systèmes de communication industriels par leur vaste gamme d'applications.

La technique d'entraînement représente l'une des principales applications dans le domaine de l'automatisation industrielle.

Les technologies ouvertes nécessitent, pour leur développement et leur diffusion sur le marché, une plateforme de travail coordonnée par une institution indépendante des entreprises. C'est dans ce but qu'a été créée, en 1989, l'Organisation des utilisateurs Profibus (PNO), une association à but non lucratif qui regroupe des constructeurs, des utilisateurs et des instituts (plus grand groupe d'intérêt mondial dans le domaine de la communication industrielle.

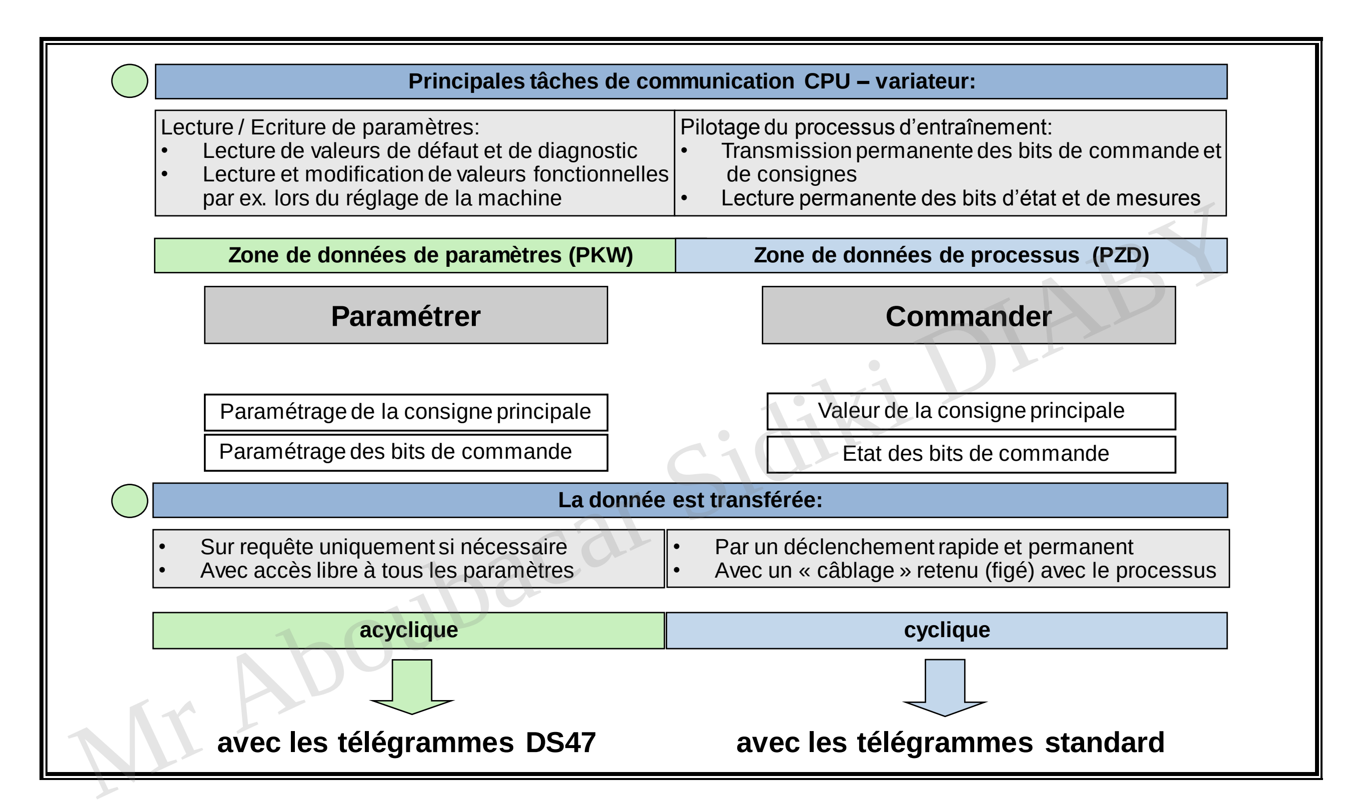
Le PNO est membre de la fédération internationale PI (PROFIBUS & PROFINET International) fondée en 1995.

PROFIdrive

PROFIdrive est le profil standard pour les systèmes d'entraînement en association avec les systèmes de communication PROFIBUS et PROFINET.
→ « Profil d'application » ouvert pour le couplage de variateurs et d'automates de différents constructeurs via des systèmes de communication


Les mots de commande et d'état du G120 sont conformes aux spécifications du profil PROFIdrive Version 2.0 ou Version 3.0 pour le mode « régulation de vitesse ».

16.2.1. Communication CPU - variateur: CPU - G120



Pilotage du processus

Il requiert un échange rapide de télégrammes courts qui doivent être distribués à l'ensemble des stations raccordées sur le réseau avec la vitesse maximale disponible sur le bus, en tenant compte que tous les participants peuvent recevoir des données différentes. Dans les systèmes d'entraînements ces données sont typiquement des consignes, des instructions de commande, des informations d'état et des valeurs réelles (valeurs mesurées).

Contrôle opérationnel

En plus de ces données permanentes, d'autres données ne sont nécessaires que dans des cas particuliers. Il serait inutile de charger le bus en permanence si ces données sont uniquement nécessaires une fois par heure ou par jour, par exemple, lors du démarrage de la machine (Maître Class 1). Une autre raison d'avoir un besoin de communication pourrait être la mise en service, l'optimisation ou le diagnostic des composants d'une machine à partir d'un poste central. Dans ce cas, (comme pour un défaut) un accès complet d'un outil d'ingénierie est possible pour ces composants du système ou du variateur. En règle générale, un seul appareil/composant est adressé directement, cette fenêtre est uniquement rendue disponible une seule fois par cycle de bus pour une seule station et non pour toutes les stations.

Echange cyclique de données:

Il sert à répondre au besoin d'un échange rapide et permanent de données.

Echange de données acyclique:

L'échange de données acycliques d'ensembles de données (par exemple DS47, DS100) entre un maître et un esclave se déroule suivant des procédures standards qui sont implémentés avec des FB issus de bibliothèques (par exemple : Drive ES SIMATIC).

L'initiative d'une transmission de données en mode acyclique pour un maître de classe 1 avec un esclave est toujours prise par le programme utilisateur. Ainsi, le programme utilisateur décide si un échange de données avec un esclave est nécessaire ou non en se basant sur d'autres conditions. C'est seulement en cas de besoin, qu'une demande d'échange de données est signalée au maitre DP et ce dernier l'exécute par la suite.

16.2.2. Télégrammes standard

Télégramme	1		20		352		354		999	
PZD1	STW1	ZSW1	STW1	ZSW1	STW1	ZSW1	STW1	ZSW1	STW1 <4>	ZSW1 <4>
PZD2	NSOLL_A	NIST_A	NSOLL_A	NIST_A_GL	NSOLL_A	NIST_A_GL	NSOLL_A	NIST_A_GL	Télégramme personnalisé de réception avec longueur libre via configuration PROFIdrive	Télégramme personnalisé d'émission avec longueur libre via configuration PROFIdrive
PZD3	Télégramme de réception PROFIBUS / PROFINET	Télégramme d'émission PROFIBUS / PROFINET		IAIST_GL	<3>	IAIST_GL	<3>	IAIST_GL		
PZD4				MIST_GL	<3>	MIST_GL	<3>	MIST_GL		
PZD5				PIST_GL	<3>	WARN_CODE	<3>	WARN_CODE		
PZD6				<2>	<3>	FAULT_CODE	<3>	FAULT_CODE		
PZD7										
PZD8										
PZD9										
PZD10										
PZD11										
PZD12										

Télégramme 1, 20	Télégrammes standard non propriétaires	Interconnexion automatique dans l'entraînement
Télégramme 352 à 391	Télégrammes standard spécifiques Siemens	Interconnexion automatique dans l'entraînement
Télégramme 999	Télégramme libre	Pas d'interconnexion automatique

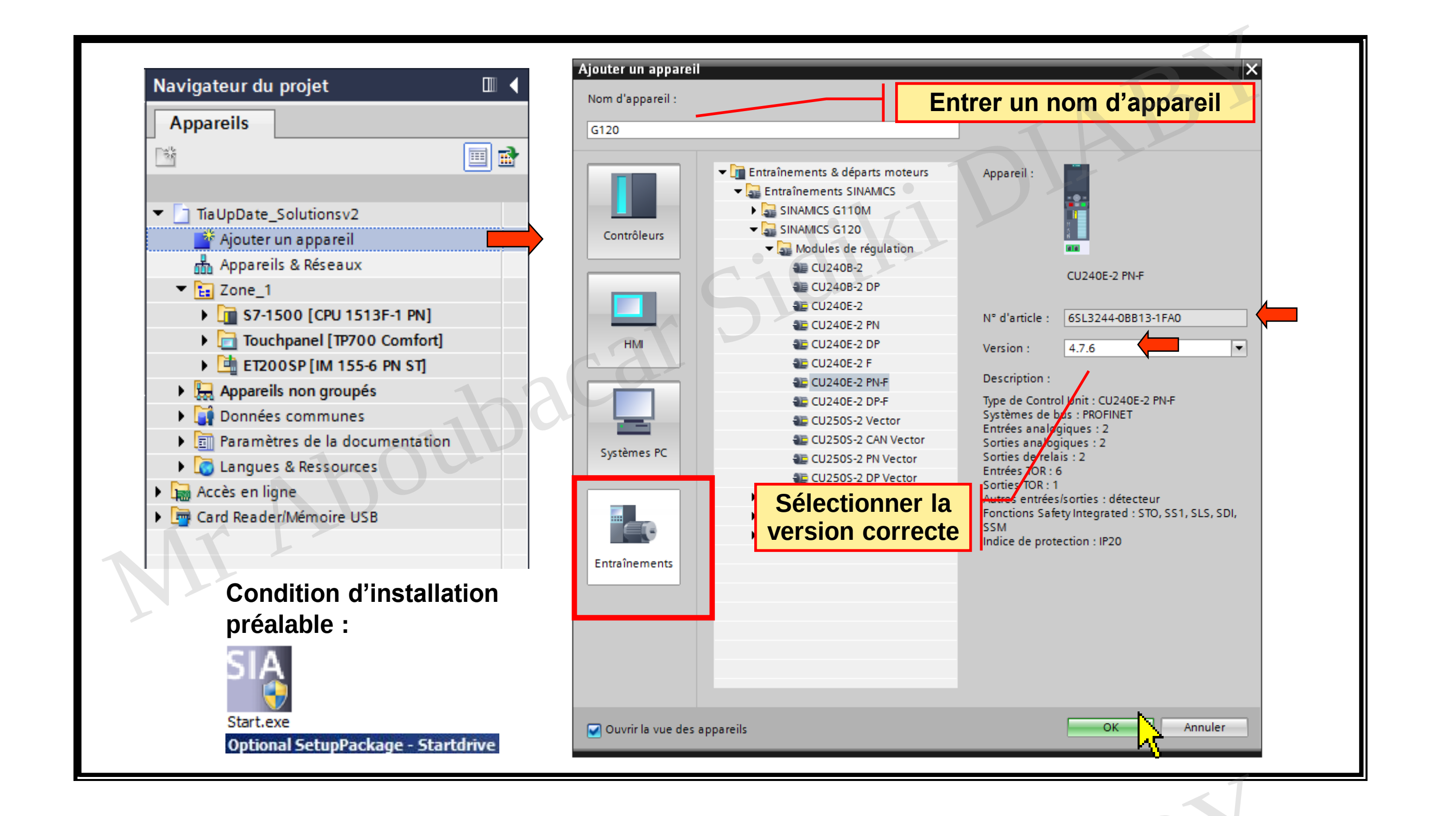
Communication Maître- Esclave / Contrôleur-Device par télégramme standard

Le meilleur moyen d'implémenter un échange cyclique de données entre un contrôleur / maître (CPU) et un Device / esclave (G120) est d'utiliser un télégramme standard, qui est aussi utilisé pour la régulation de vitesse simple. Entre CPU et G120, deux mots d'entrée et deux mots de sortie avec le contenu suivant sont alors échangés:

- Mot de sortie 1: Mot de commande
- Mot de sortie 2: Mot de consigne principale
- Mot d'entrée 1: Mot d'état
- Mot d'entrée 2: Mot pour mesure principale

Mr Aboubacar Sidiki DIABY

16.3. Ajouter un variateur dans le projet

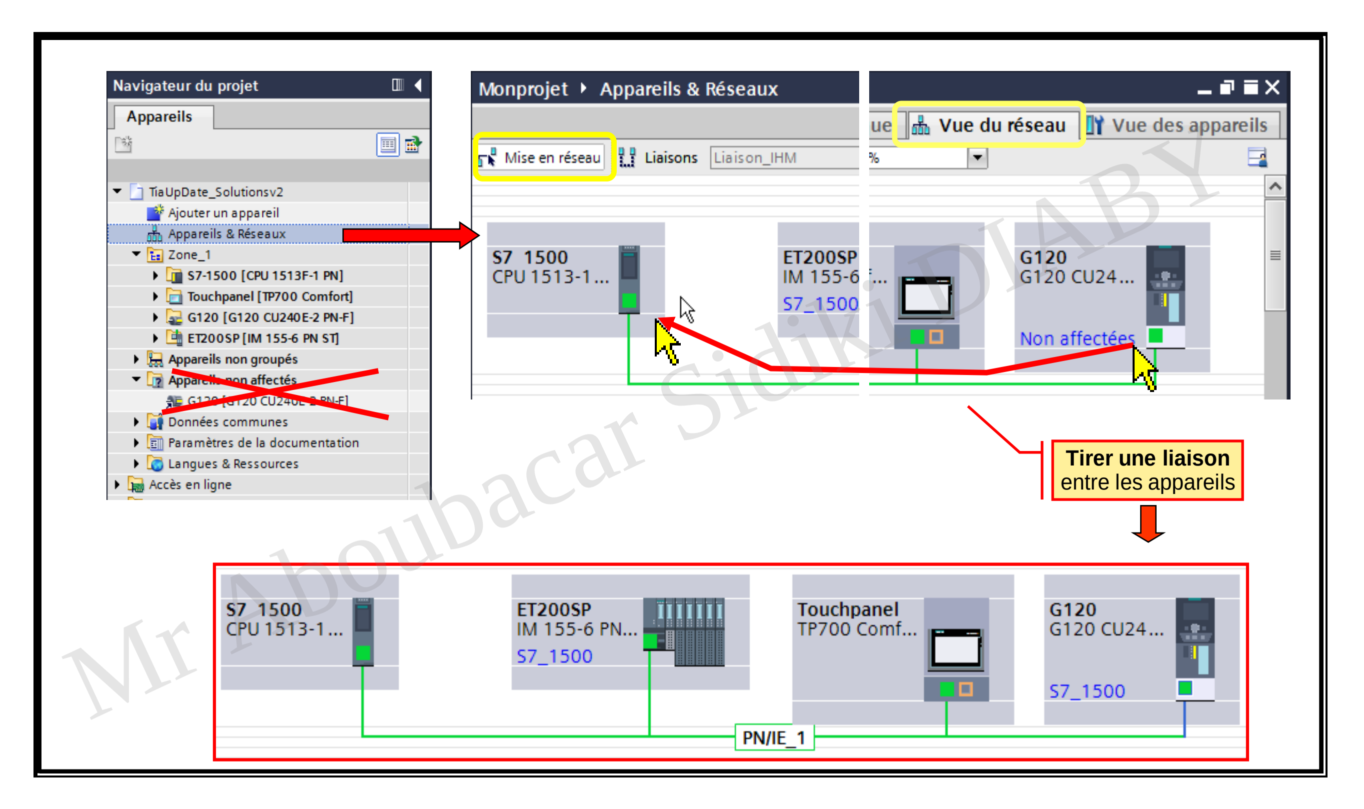


Pack optionnel « Startdrive »

Si le pack optionnel « Startdrive » est installé dans le TIA Portal, il est possible d'ajouter un variateur comme nouvel appareil puis de le configurer et de le paramétrer.

Sinon, le variateur peut être ajouté dans le projet avec le fichier GSD comme station périphérique décentralisée à partir du catalogue du matériel.
Cependant, uniquement la communication avec la CPU peut être configurée avec TIA Portal, pas de configuration matérielle ni de paramétrage.

16.3.1. Mettre le variateur en réseau avec la CPU



Editeur

« Appareils & Réseaux » → « Vue du réseau »

Mise en réseau

Cliquer sur l'interface d'un des modules, maintenir le bouton gauche de la souris enfoncé et tirer une liaison avec l'interface de l'autre module.
→ dans la figure ci-dessus : liaison entre l'interface PN du G120 et l'interface PN de la CPU

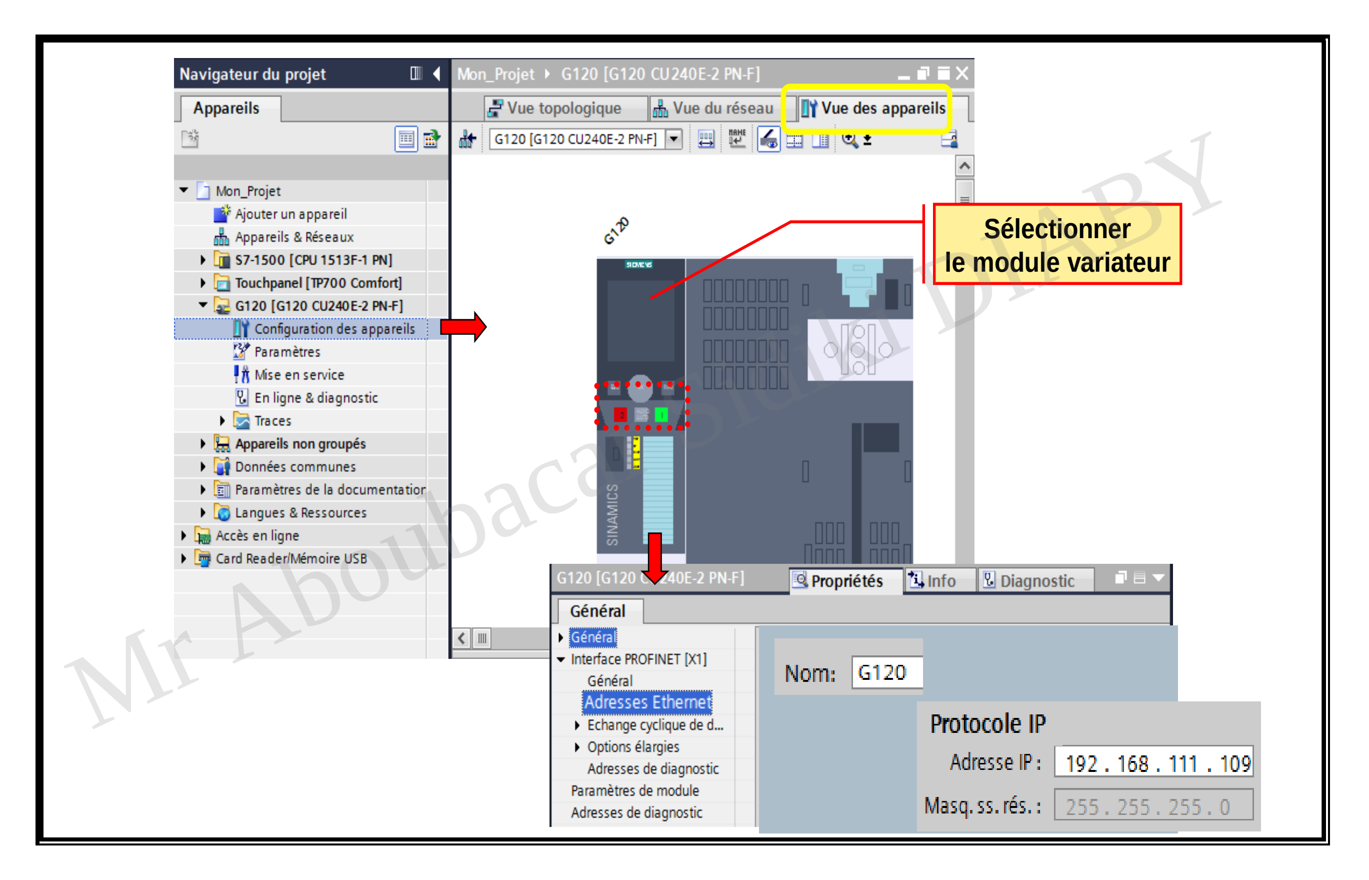
Appareils non affectés

Cette rubrique disparaît alors puisque l'appareil est à présent affecté à la CPU.

Le variateur apparaît en périphérie décentralisée



16.3.2. Paramétrer l'adresse et le nom du module



Nom du module

Le nom est l'affectation CPU ↔ Module.

Il permet au contrôleur PROFINET d'identifier le PROFINET IO.

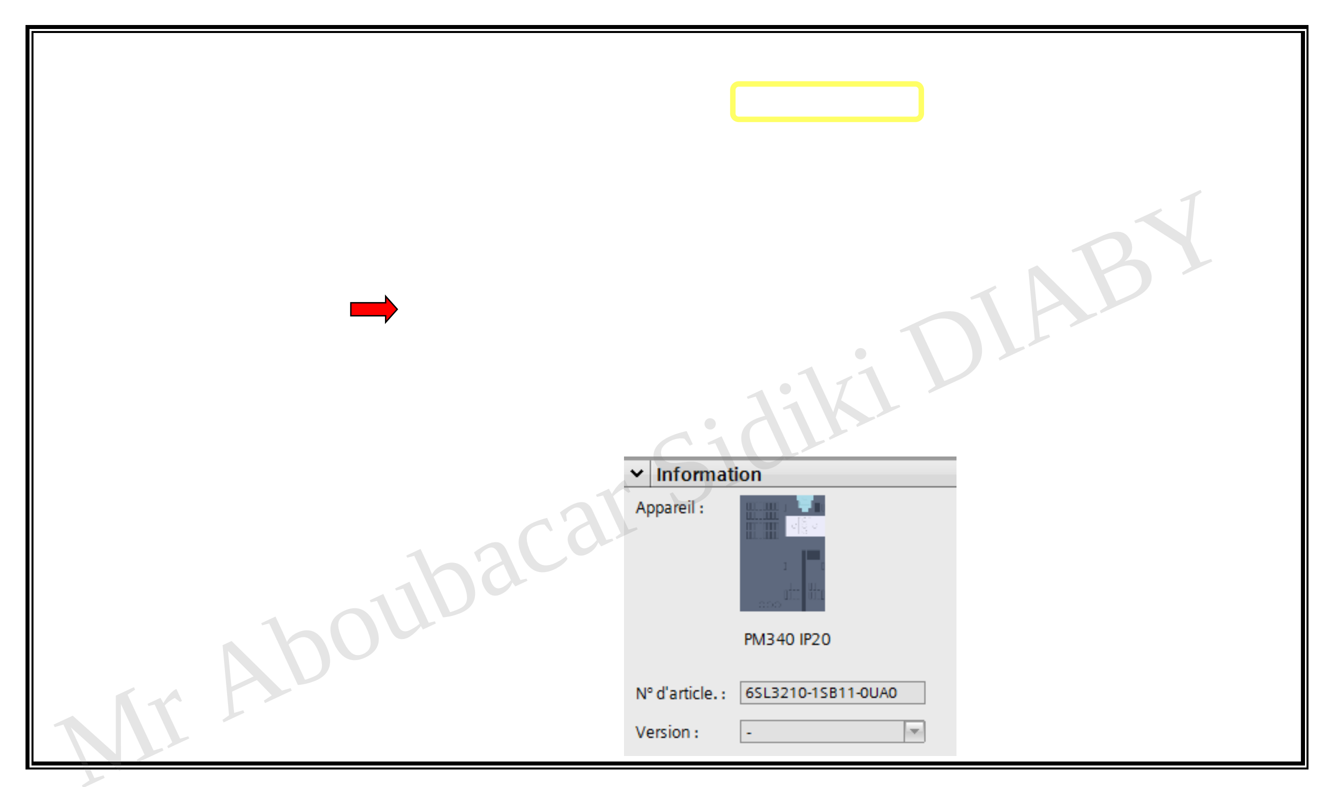


Ce nom doit être affecté en ligne au module PROFINET IO, sinon la CPU (contrôleur PROFINET-I/O) ne peut pas trouver le module.

Affectation d'adresse au module IO décentralisé

Avec PROFINET IO, l'adresse IP paramétrée du module est affectée par la CPU via le nom d'appareil configuré.

16.3.3. Configurer la partie puissance

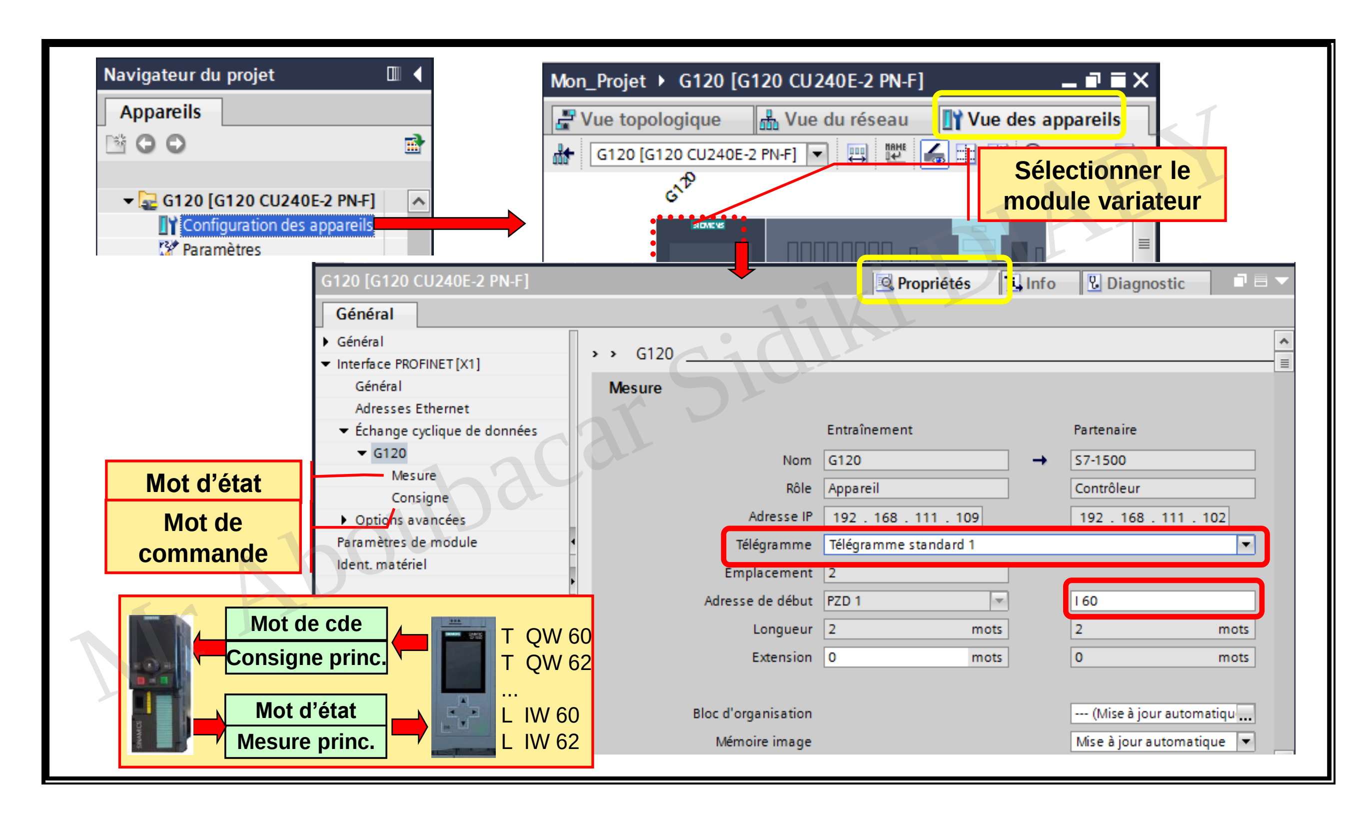


Configurer la partie puissance

Un variateur se compose de deux parties :

- Module de régulation → déjà configuré avec l'appareil ajouté
- Partie puissance → doit être configurée

16.3.4. Paramétrer la zone de données processus (PZD)



Zone de données d'échange

La communication via le télégramme standard 1 est un échange de données cyclique très simple à programmer.

 Les mots de commande et d'état du G120 sont conformes aux spécifications du profil PROFIdrive Version 2.0 ou Version 3.0 pour le mode « régulation de vitesse ».

Le mot de commande et, si nécessaire, la consigne principale sont envoyés par la CPU au variateur. Le variateur renvoie le mot d'état et la mesure principale dans le télégramme de réponse.

Comme deux mots seulement sont échangés entre le variateur et la CPU, il suffit de programmer de simples opérations de chargement et de transfert. La cohérence des données est également garantie par le transfert d'un double-mot.

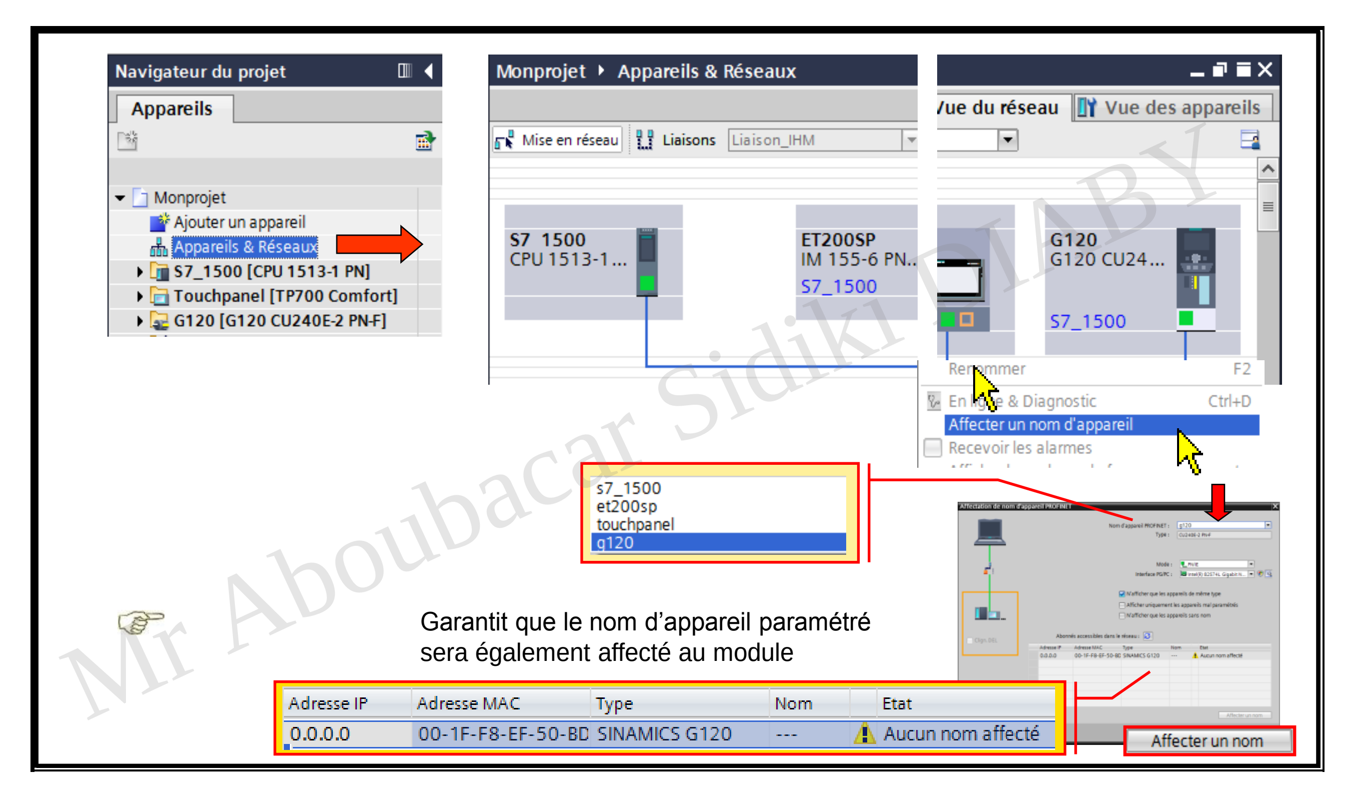
Mot de commande

Le mot de commande contient 16 signaux binaires pour la commande (Marche/Arrêt, sens de rotation) du variateur.

Consigne principale

Selon le paramétrage standard du variateur, la consigne principale spécifie la vitesse de consigne.

16.3.5. Affecter un nom d'appareil (nommage du module)



Le nom PROFINET du module Hors ligne doit être attribué à l'entraînement En ligne en place.

Nom du module

Le nom est l'affectation CPU ↔ Module.

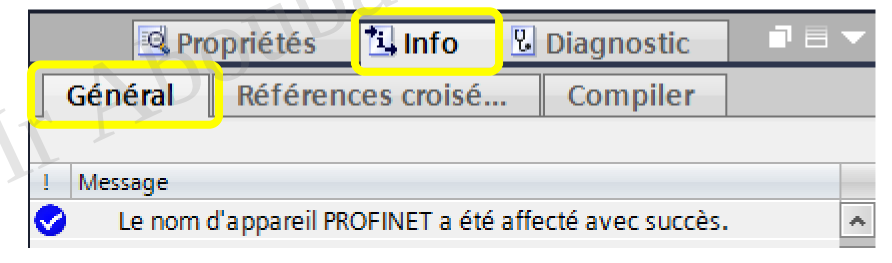
Il permet au contrôleur PROFINET d'identifier le PROFINET IO.

 **Ce nom doit être affecté en ligne au module PROFINET IO, sinon la CPU (contrôleur PROFINET) ne peut pas trouver le module.**

Affectation du nom

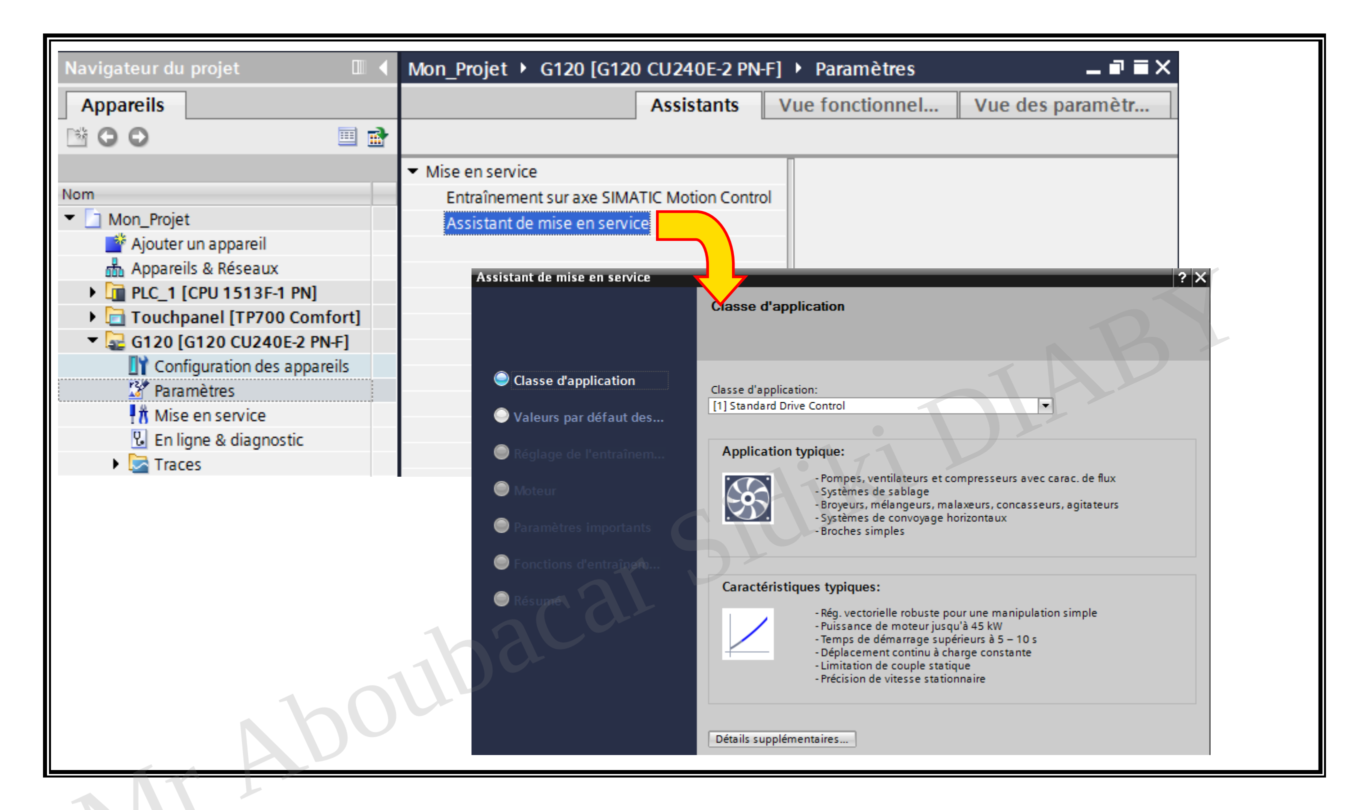
Actuellement, cette option de menu n'est disponible que dans le **menu contextuel de la liaison réseau**, et non pour le variateur.

Vérifier le résultat dans la fenêtre d'inspection



Remarque :

Si le nom de l'appareil n'est pas affecté via le dialogue « Affecter un nom d'appareil » mais avec l'accès en ligne : « En ligne & Diagnostic/Fonction/Affectation d'un nom d'appareil » et si l'appareil dispose déjà d'un nom alors il faut redémarrer l'entrainement.



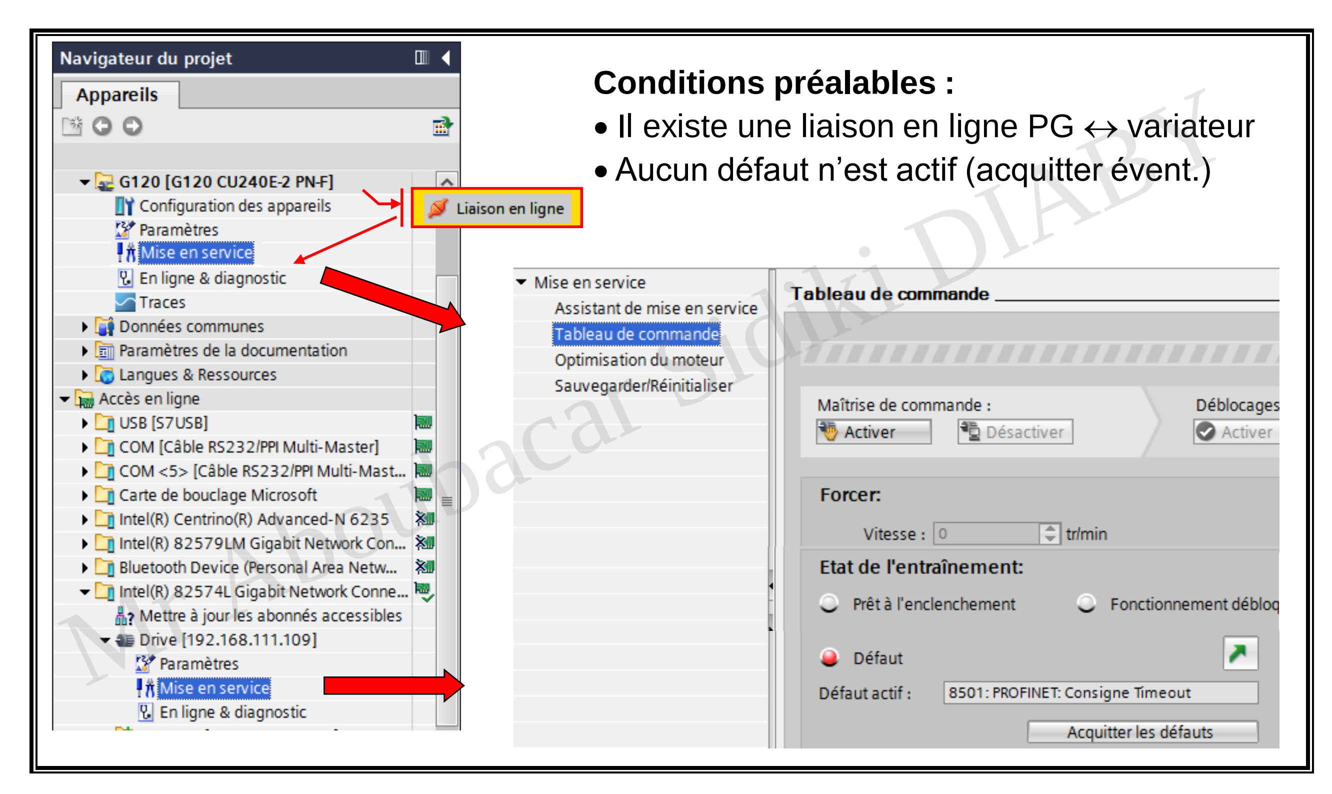
L'assistant de mise en service

Les réglages les plus importants pour le variateur peuvent être réalisés avec les menus de l'assistant de mise en service. Ils concernent:

- Les Data sets
- Les paramètres de la boucle de régulation ouverte ou fermée
- Réglages par défaut des consignes et des commandes
- Les paramétrages du variateur
- Les caractéristiques du moteur (données)
- Les paramètres importants (données dynamiques)
- Les fonctions du variateur

16.4. Tableau de commande pour la mise en service

16.4.1. Activer/désactiver le tableau de commande



Le paramétrage et les fonctions de mise en service sont disponible dès que le variateur possède une adresse IP valide et est, par conséquent, accessible en ligne.

Accès en ligne

Dès qu'une adresse IP valide dans le sous-réseau utilisé a été affectée au variateur, les fonctions de mise en service sont disponibles ici.

Appareil dans le navigateur du projet



Si le variateur a été ajouté comme nouvel appareil, l'adresse IP par défaut 192.168.0.1 lui est affectée.

Liaison en ligne n'est possible que si l'adresse IP de la configuration de l'appareil correspond au sous-réseau utilisé et est également affectée au variateur.

Tableau de commande



Le tableau de commande ne peut être activé que si une liaison en ligne est établie et qu'aucun défaut n'est présent.



Attention !
L'activation interrompt une liaison éventuellement établie avec la CPU et donne à l'opérateur la totale maîtrise de commande sur le variateur. Il est de ce fait important de disposer d'une solution matérielle pour un arrêt d'urgence éventuel.

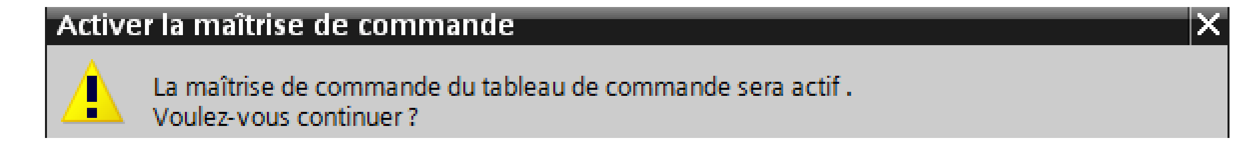
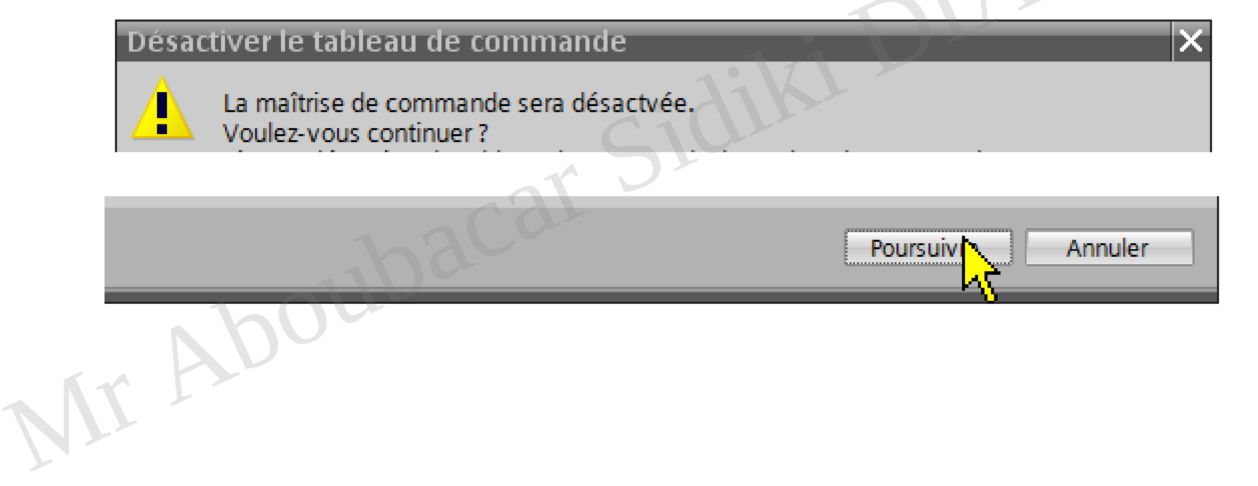




Tableau de commande 

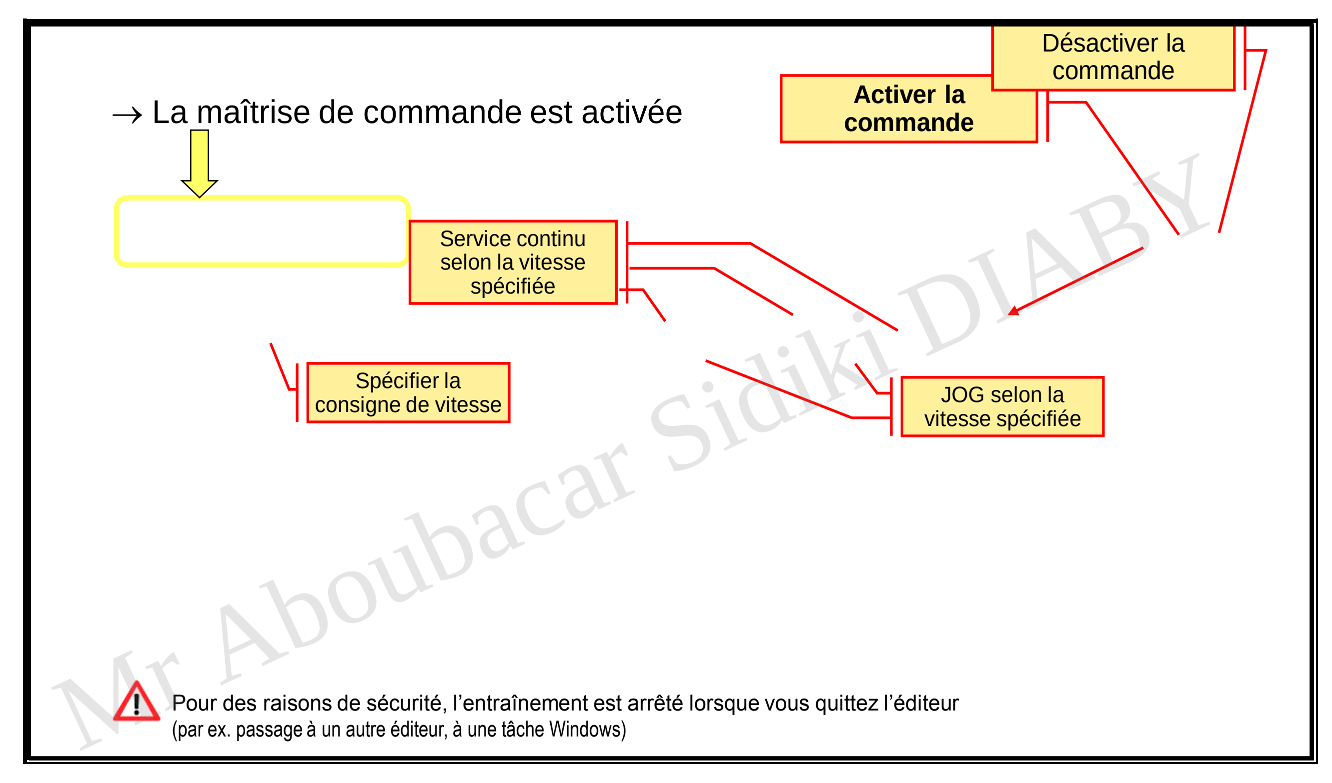


Attention !
La désactivation de la maîtrise de commande du tableau de commande active une liaison éventuellement établie avec la CPU et redonne à la CPU la totale maîtrise de commande sur le variateur. Les commandes de commutation de la CPU éventuellement présentes redeviennent immédiatement actives.



Mr Aboubacar Sidiki DIABY

16.4.2. Commander le tableau de commande



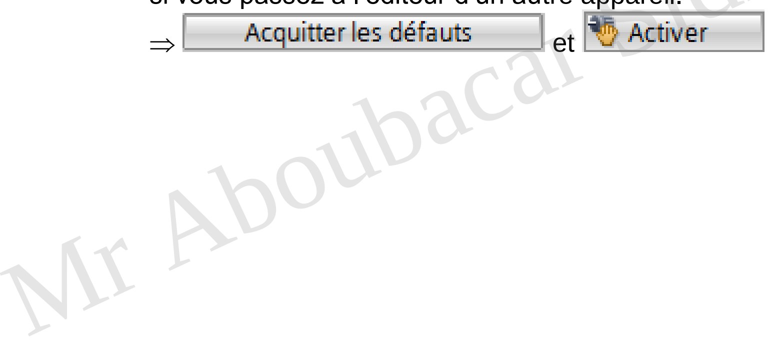
Si la maîtrise de commande s'effectue au niveau de la PG, le forçage peut être activé.

Le fonctionnement est validé et les boutons tactiles sont actifs pour la commande.

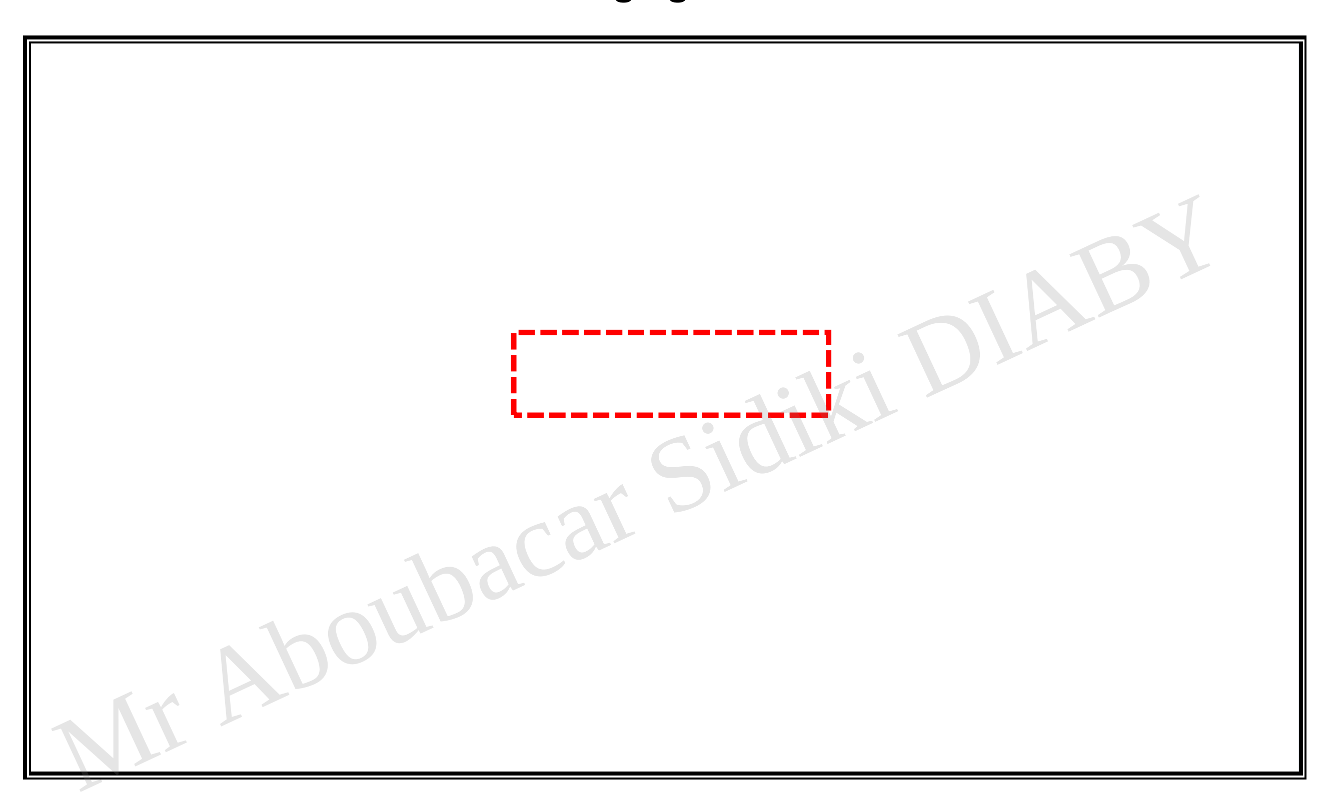
Mise à l'arrêt de sécurité

L'entraînement peut être arrêté et...

- les validations de fonctionnement sont désactivées
  si vous quittez l'éditeur « Mise en service » en basculant sur une autre tâche Windows.
  ⇒ Activer
- la maîtrise de commande est désactivée avec affichage d'un message d'erreur
  si vous passez à l'éditeur d'un autre appareil.
  ⇒ Acquitter les défauts et Activer



16.5. Exercice 1: restaurer le réglage d’usine



Enoncé

Le variateur doit être réinitialisé aux réglages d’usine.

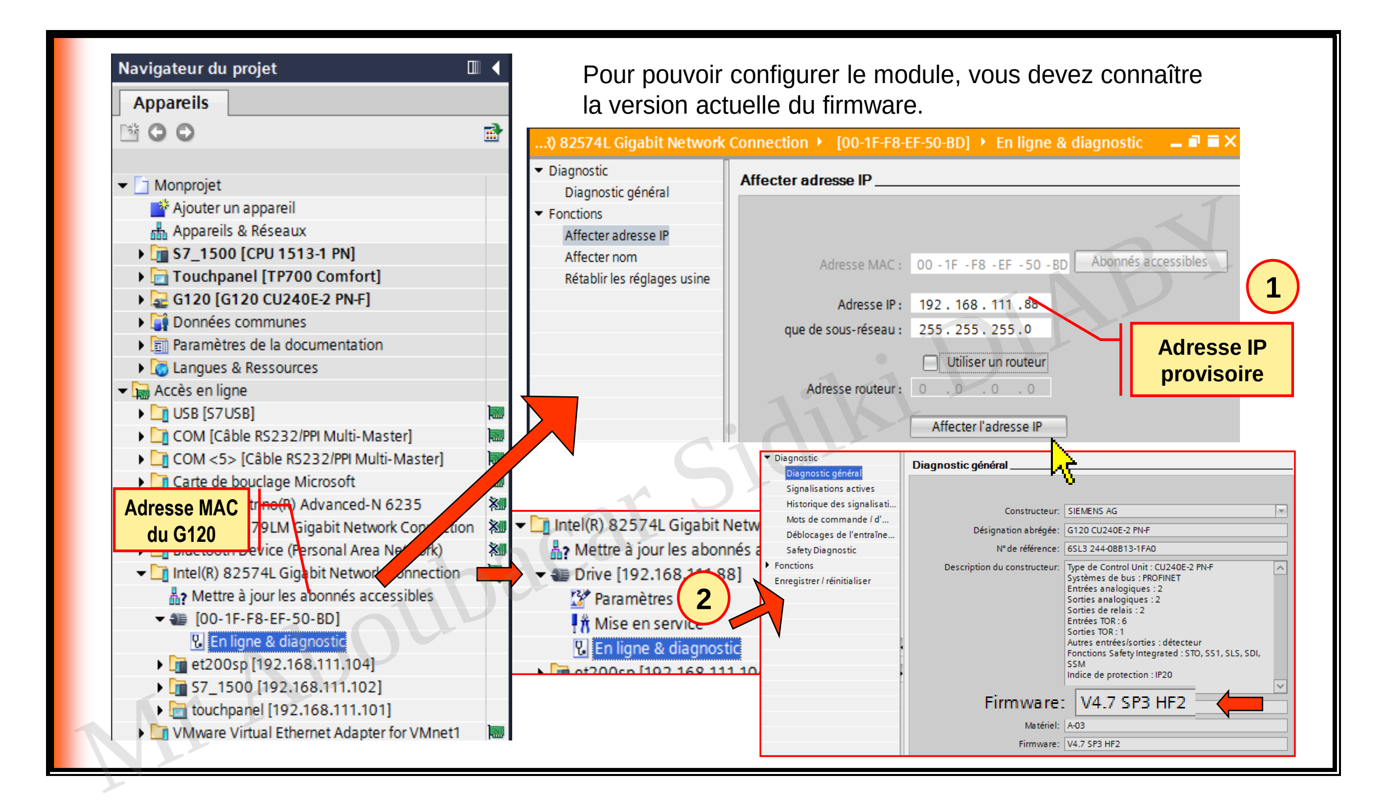
Avec la fonction « rétablir le réglage d’usine » le nom d’appareil PROFINET et l’adresse IP sont effacés.

Marche à suivre

1. Retenez parmi les accès en ligne l’interface Intel(R) 82574L Gigabit....et actualisez la liste des abonnés accessibles.
2. Ouvrez le variateur et sous « En ligne & diagnostic »
3. Activez dans la fenêtre de droite la fonction « Rétablir les réglages usine » et lancer la fonction « réinitialiser ».
4. Refermez la fenêtre.

Remarque : il peut être nécessaire d’affecter une adresse IP au variateur pour accéder à la fonction de réinitialisation.

16.5.1. Exercice 2 : lire la version du firmware du variateur



Enoncé

Pour pouvoir configurer le variateur dans le projet des exercices suivants, vous devez connaître la version du firmware de l'appareil utilisé pour la formation.
Comme le firmware du variateur est téléchargeable, vous ne pouvez pas être certain de la version actuellement utilisée. Vous devez donc lire en ligne la version actuelle du firmware dans la CU.

Indiquez la version du firmware ainsi déterminée dans la vue d'ensemble de l'exercice 1 du chapitre 1.

Problème

Du fait du précédent « rétablissement des réglages usine », le variateur n'a plus de nom d'appareil PROFINET ni d'adresse IP. Il ne dispose plus que d'une adresse MAC (voir figure ci-dessus). L'adresse MAC ne permet toutefois pas de lire la version du firmware, car une adresse IP est exigée pour ce service de diagnostic.

Solution

1. Affectez une adresse IP provisoire qui soit adaptée au sous-réseau actuel et qui ne soit pas encore utilisée (par ex. 192.168.111.88).
2. Lisez la version du firmware et notez-la

Marche à suivre

1. Ouvrir « En ligne & Diagnostic » du variateur

Ouvrez le variateur et activez par un double-clic la fonction « En ligne & Diagnostic »".



Si l'exercice 1 a été correctement effectué, l'adresse MAC est à présent nécessaire pour l'identification. Celle-ci est imprimée sur le boîtier de la CU (ControlUnit) du variateur.

Aucune information sur le module ne doit être affichée sous l'onglet « Diagnostic général »….

...twork Connection ▸ [00-1F-F8-EF-50-BD] ▸ En ligne & diagnostic

▾ Diagnostic
Diagnostic général
▸ Fonctions

Diagnostic général

Données d'identification et de maintenance:

Constructeur:
N° de réf.:
Numéro de série:
Matériel:
Firmware:

...sinon, recommencez l'exercice 1 car le module n'a pas été réinitialisé.

2. Affecter une adresse IP provisoire au variateur

Dans la fenêtre de travail de « En ligne & Diagnostic », ouvrez la rubrique « Fonctions », puis sélectionnez « Affecter l'adresse IP ».

Entrez ici l'adresse IP provisoire indiquée sur la figure ainsi que le masque de sous-réseau, puis activez le transfert de l'affectation via Affecter l'adresse IP

Vérifiez le résultat dans la fenêtre d'inspection sous l'onglet « Info »+ »Général ».



Les paramètres ont été transférés avec succès.

3. Lire la version du firmware

Dans le navigateur du projet, activez une nouvelle fois « Mettre à jour les abonnés accessibles ».

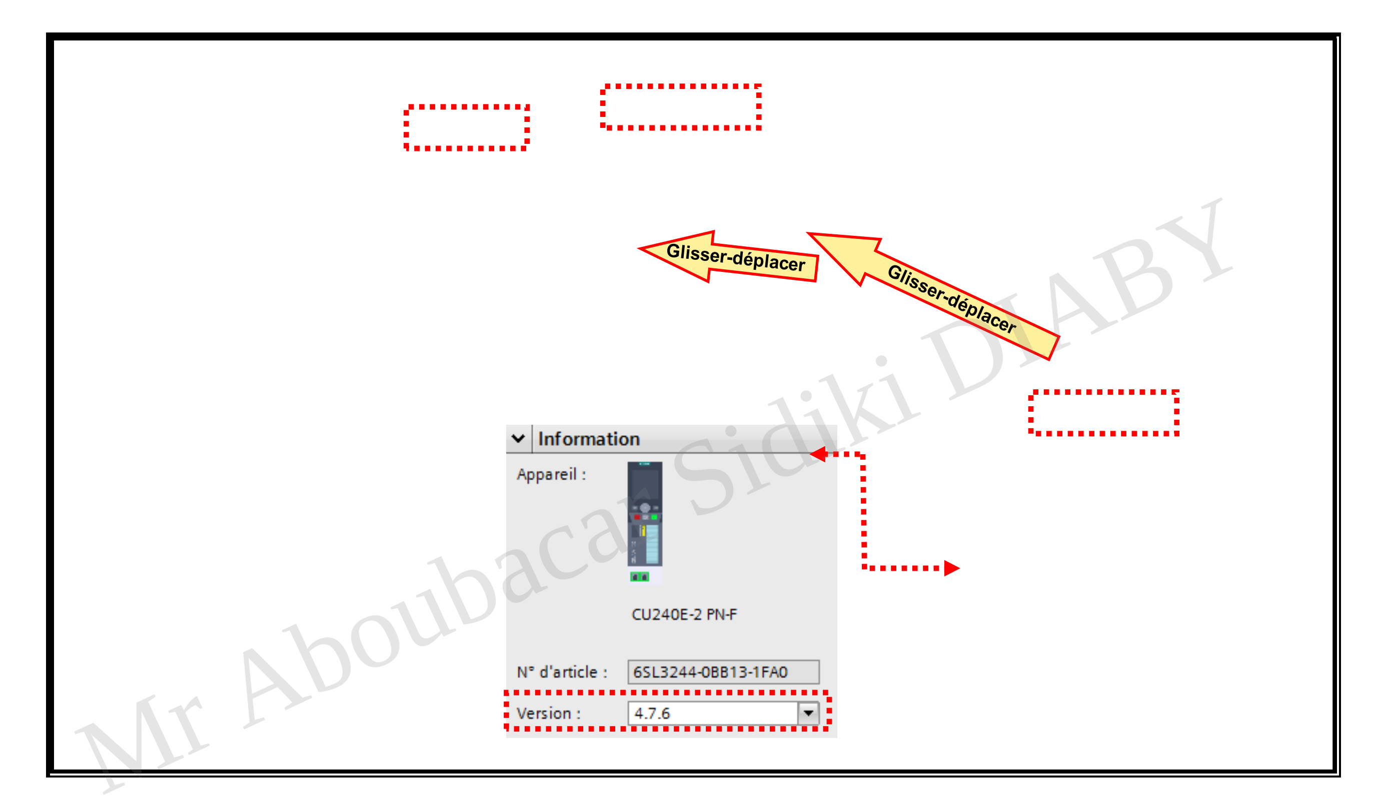
Le variateur s'affiche alors dans la liste des abonnés avec l'adresse IP affectée. Activez une nouvelle fois « En ligne & Diagnostic ».

Notez la version du firmware affichée sous l'onglet « Diagnostic -> Général ».

4. Fermer « En ligne & Diagnostic »

Fermez l'éditeur « En ligne & Diagnostic »

16.5.2. Exercice 3: Insertion du variateur dans le réseau du projet Hors Ligne



Enoncé

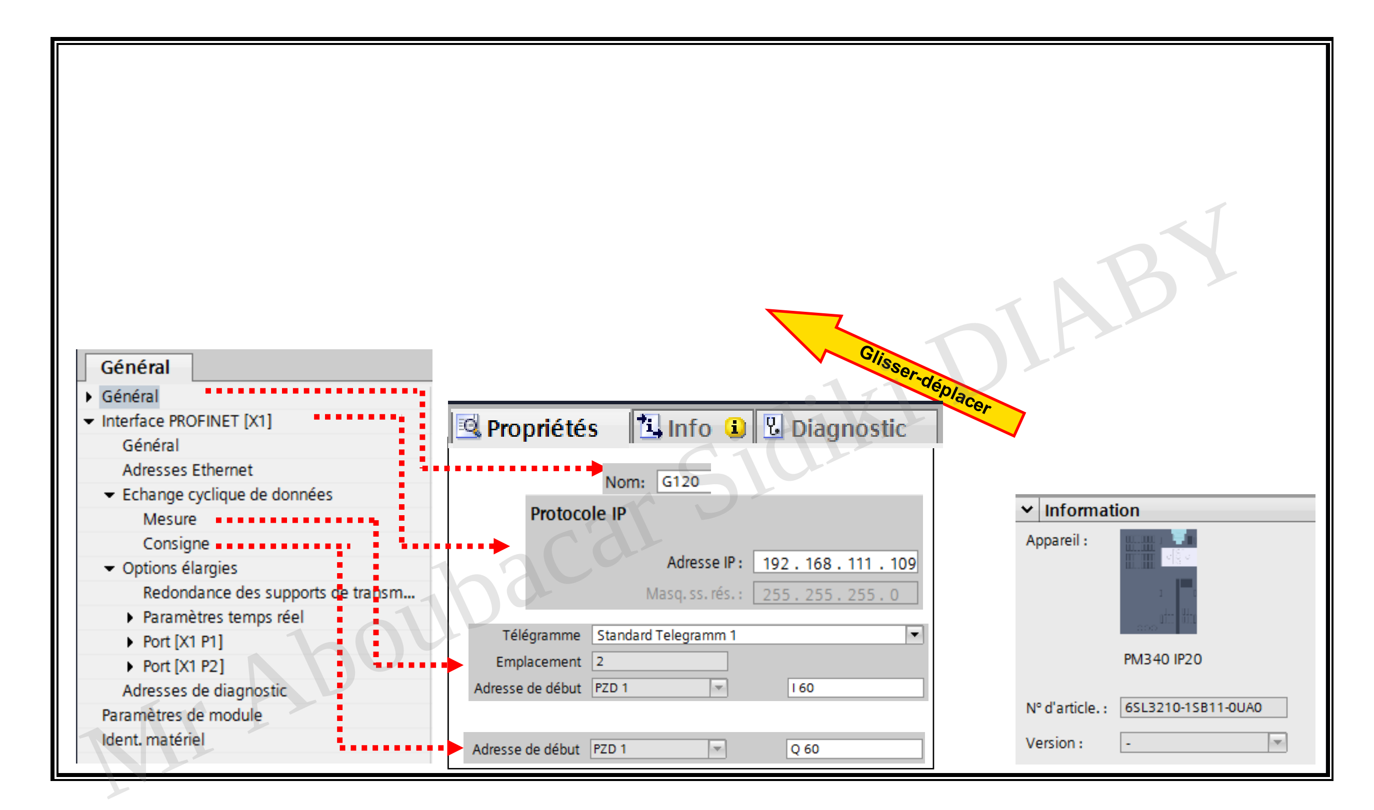
Le variateur doit être inséré dans le projet et relié à la CPU via le réseau PROFINET.

Marche à suivre

1. Ouvrez l'éditeur « Appareils & Réseaux » et activez la « Vue du réseau ».
2. ouvrez dans la « Task-Card » le catalogue matériel
3 Insérez le G120 par glisser – déposer du module de régulation CU240E-2PN-F dans la vue du réseau.

   Faîtes attention à la référence et à la version sous « Information ».
4. Raccordez le variateur à la CPU par glisser- déposer de l'interface PROFINET du variateur sur l'interface PROFINET de la CPU.
5. Sauvegardez votre projet.

16.5.3. Exercice 4: Paramétrer la communication du variateur



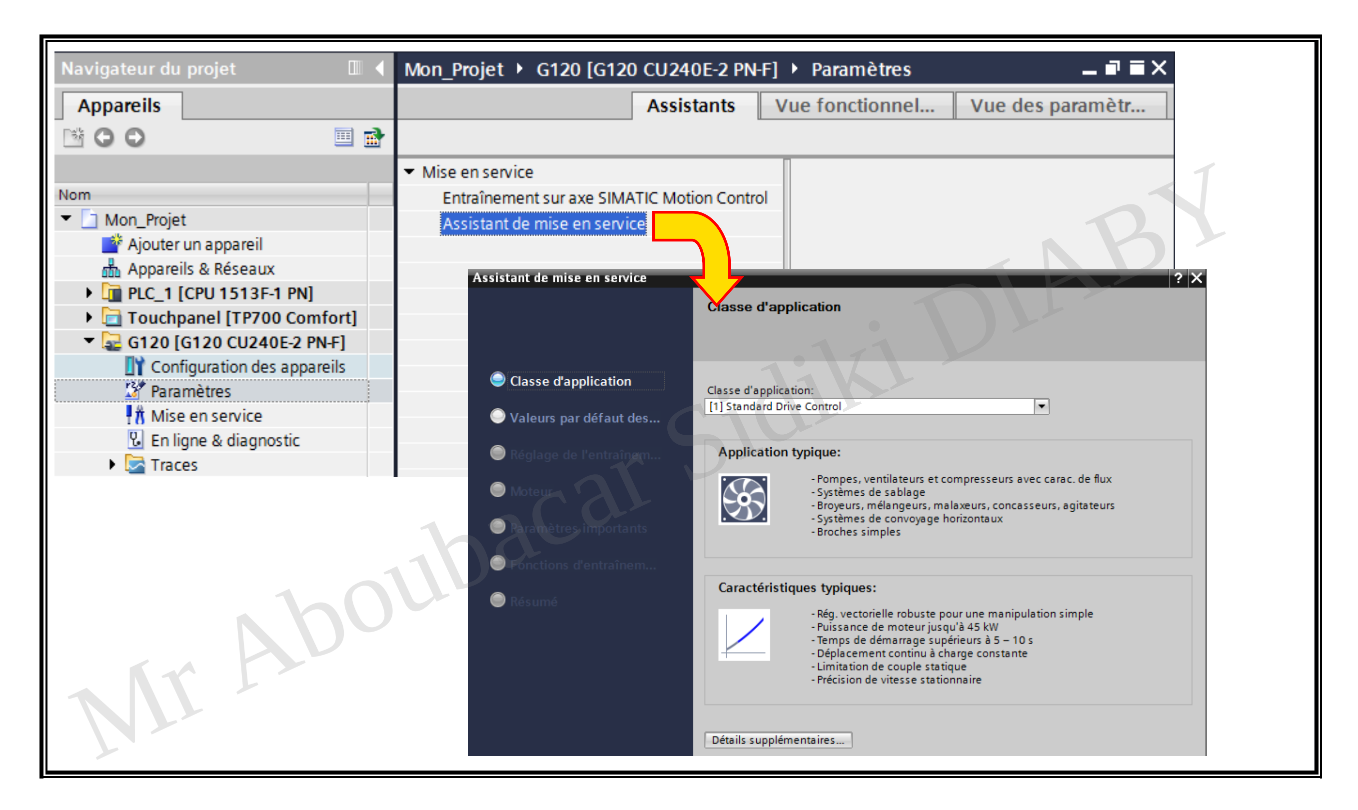
Enoncé

Pour compléter la configuration de l'entraînement, il faut lui adjoindre une partie puissance. Par la suite il faudra encore paramétrer la partie régulation.

Mise en œuvre:

1. ouvrez la configuration matérielle de l'entraînement.
2. ouvrez le catalogue matériel et insérez la partie puissance par glisser-déposer. Faîtes attention à la référence et à la version sous « Information ».
3. Sélectionnez le module de régulation et paramétrez le dans la fenêtre d'inspection sous « propriétés » comme indiqué dans la figure ci-dessus.
4. Sauvegardez votre projet.

16.5.4. Exercice 5: Paramétrage Hors Ligne du variateur avec l'assistant de mise en service



Enoncé

Réglez les principaux paramètres du variateur.

Le plus simple consiste à utiliser l'assistant de mise en service.

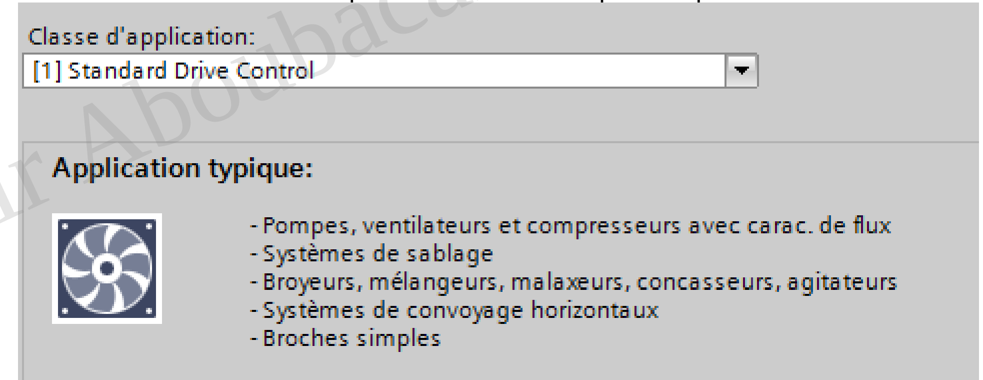
Marche à suivre

1. Lancer l'assistant de mise en service

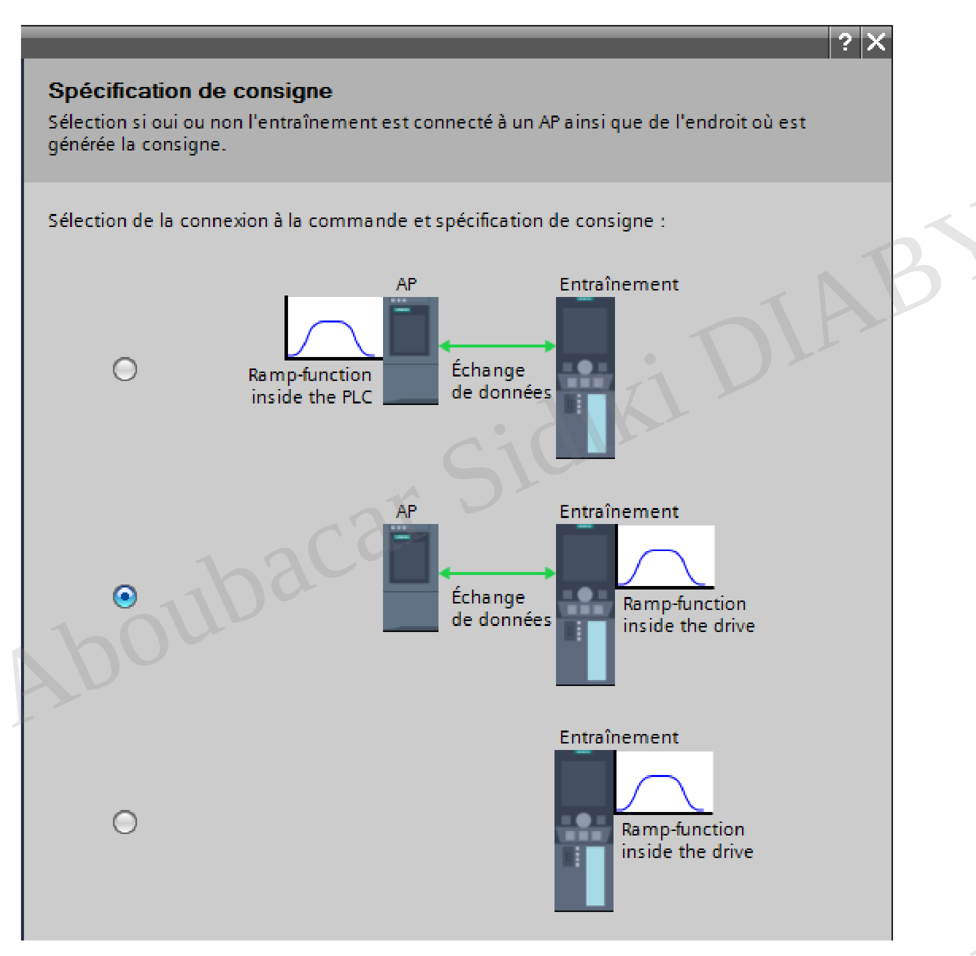
Ouvrez l'éditeur « Mise en service » et lancez l'assistant de mise en service.

2. Type de commande/régulation

Une commande tension-fréquence est suffisante pour ce petit moteur.



3. Valeurs par défaut des consignes et sources de commande



Validez les réglages par défaut.

Configuration E/S actuelle :

[7] Bus de terrain avec commutation entre jeux de paramètres

Sélectionnez la configuration E/S par défaut:

Aucune modification

Remarque : lors de la modification, toutes les connexions internes à l'entraînement existantes sont supprimées sur les bornes E/S et reconnectées selon la configuration E/S sélectionnée.

Appliquer

Connexions actuelles des bornes E/S :

DI 0: p1055[1] BI: JOG Bit 0

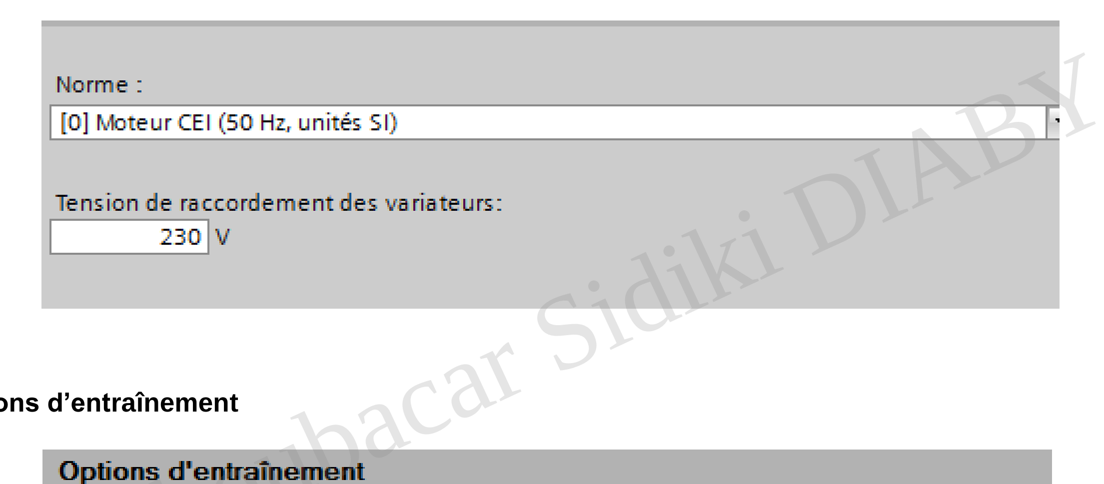
DI 1: p1056[1] BI: JOG Bit 1

Configuration de télégramme actuelle :

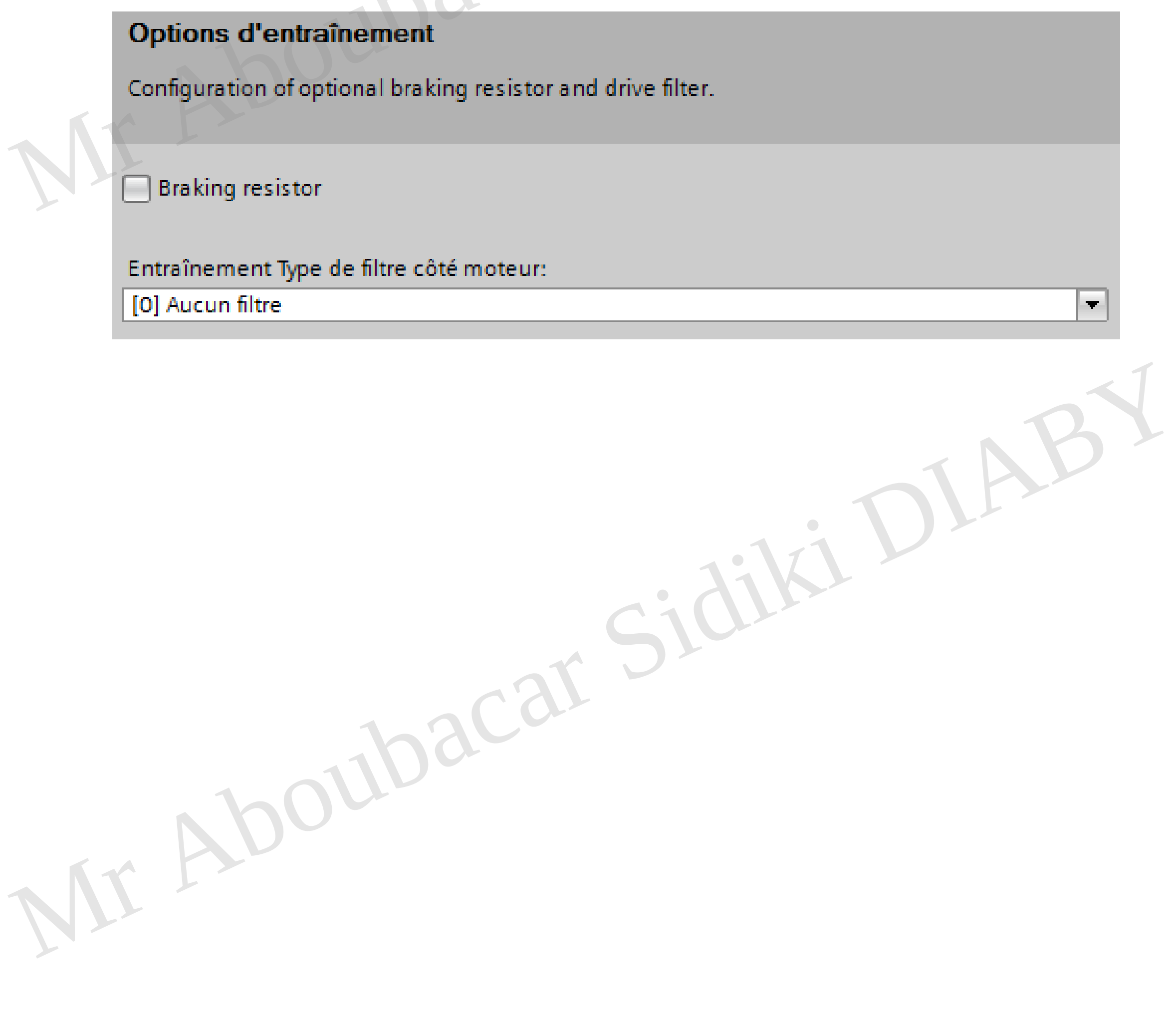
[1] Télégramme standard 1, PZD-2/2

4. Réglage de l'entraînement

Validez les réglages par défaut.



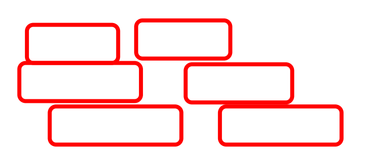
5. Options d'entraînement



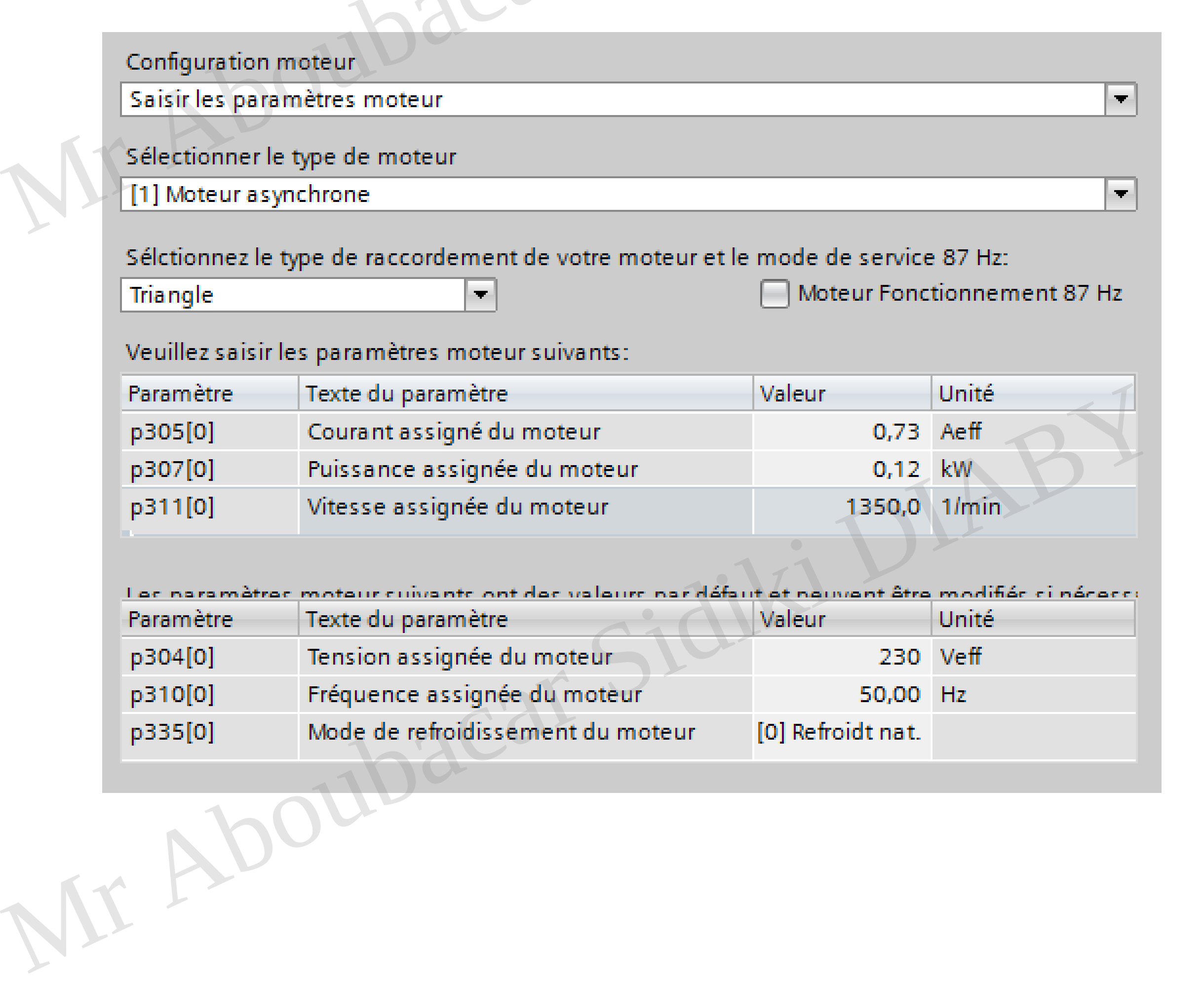
Mr Aboubacar Sidiki DIABY

6. Moteur

Pour renseigner les paramètres du moteur, vous devez récupérer les indications sur la plaque signalétique du moteur par ex.



Ces informations figurent également dans la liste des types de moteur, si celui-ci est proposé.



Configuration moteur

Saisir les paramètres moteur

Sélectionner le type de moteur

[1] Moteur asynchrone

Sélctionnez le type de raccordement de votre moteur et le mode de service 87 Hz:

Triangle

Moteur Fonctionnement 87 Hz

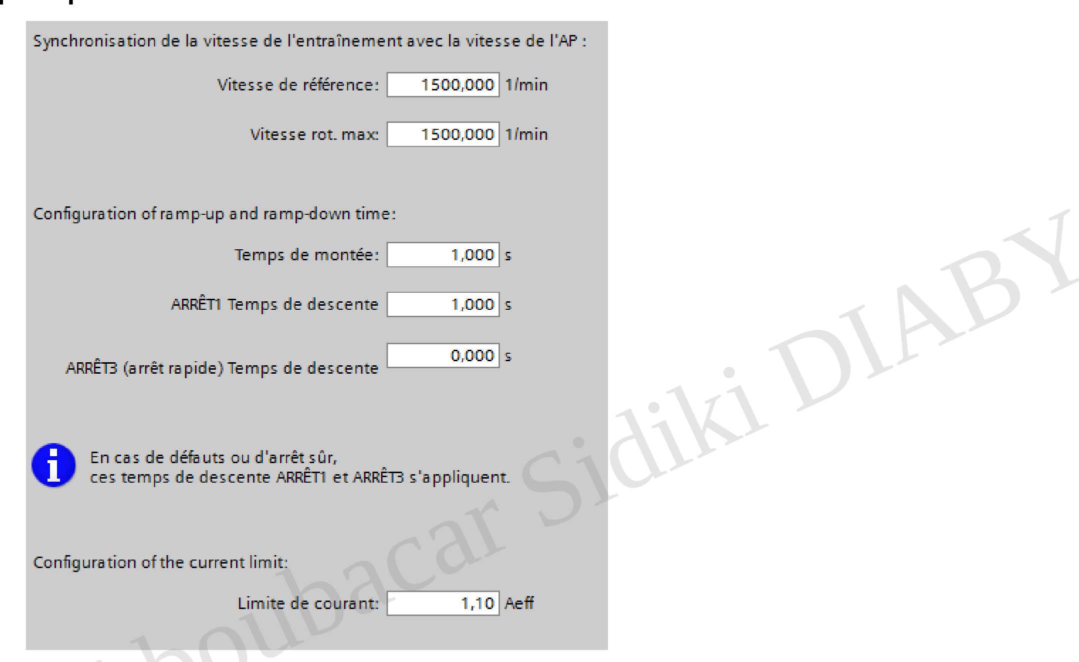
Veuillez saisir les paramètres moteur suivants:

Paramètre	Texte du paramètre	Valeur	Unité
p305[0]	Courant assigné du moteur	0,73	Aeff
p307[0]	Puissance assignée du moteur	0,12	kW
p311[0]	Vitesse assignée du moteur	1350,0	1/min

Les paramètres moteur suivants ont des valeurs par défaut et peuvent être modifiés si nécess:

Paramètre	Texte du paramètre	Valeur	Unité
p304[0]	Tension assignée du moteur	230	Veff
p310[0]	Fréquence assignée du moteur	50,00	Hz
p335[0]	Mode de refroidissement du moteur	[0] Refroidt nat.	

7. Principaux paramètres

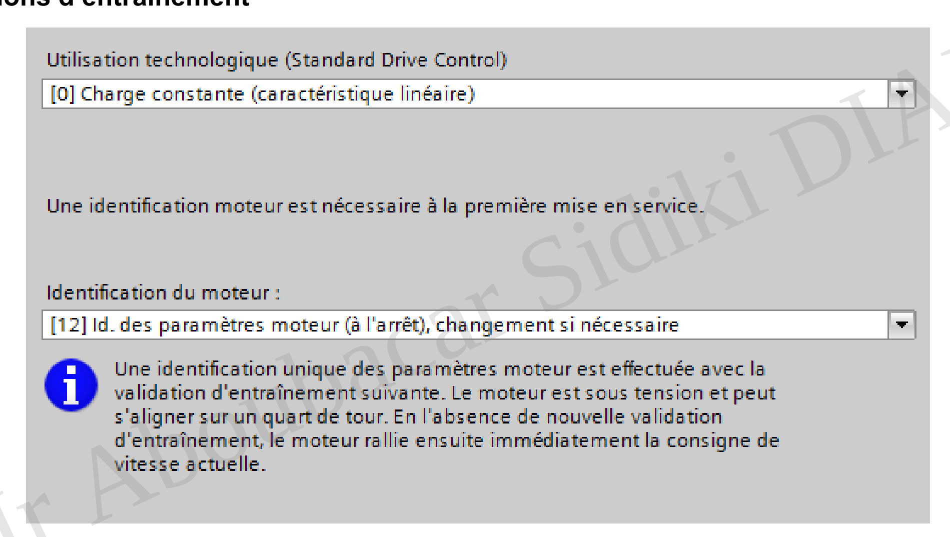


La limite de courant proposée par le variateur est 1,10Aeff. Vous pouvez reprendre cette valeur. Valeur indicative : env. 2 x courant assigné.

Comme vitesse maximale, vous ne devez pas entrer ici plus de 2500 tr/min, car ce petit moteur ne doit pas subir de contraintes mécaniques trop importantes.


Comme la consigne/mesure est convertie sur le pupitre opérateur, vous devez sélectionner la base de calcul 1500 tr/min pour éviter que le pupitre opérateur n'affiche des valeurs de vitesse fausses.

8. Fonctions d'entraînement



- Rétablir à nouveau les réglages usine
  afin de réinitialiser TOUTES les données du moteur contenues dans le variateur.
- Calculer les paramètres moteur :
  Les données moteur nécessaires doivent être calculés à partir des paramètres (points 6 et 7) et mémorisés dans le variateur.


Une identification du moteur (pour un calcul plus précis des données moteur calculées) est recommandée dans la pratique, mais n'est pas nécessaire pour notre exercice. Ne traitez pas cette fonction.

9. Résumé de la configuration

Les données de l'entraînement suivantes sont saisies :

Classe d'application:
Classe d'applications: [1] Standard Drive Control

Valeurs par défaut des sources de consigne / de commande:
Macro Groupe d'entraînement: [7] Bus de terrain avec commutation entre jeux de paramètres

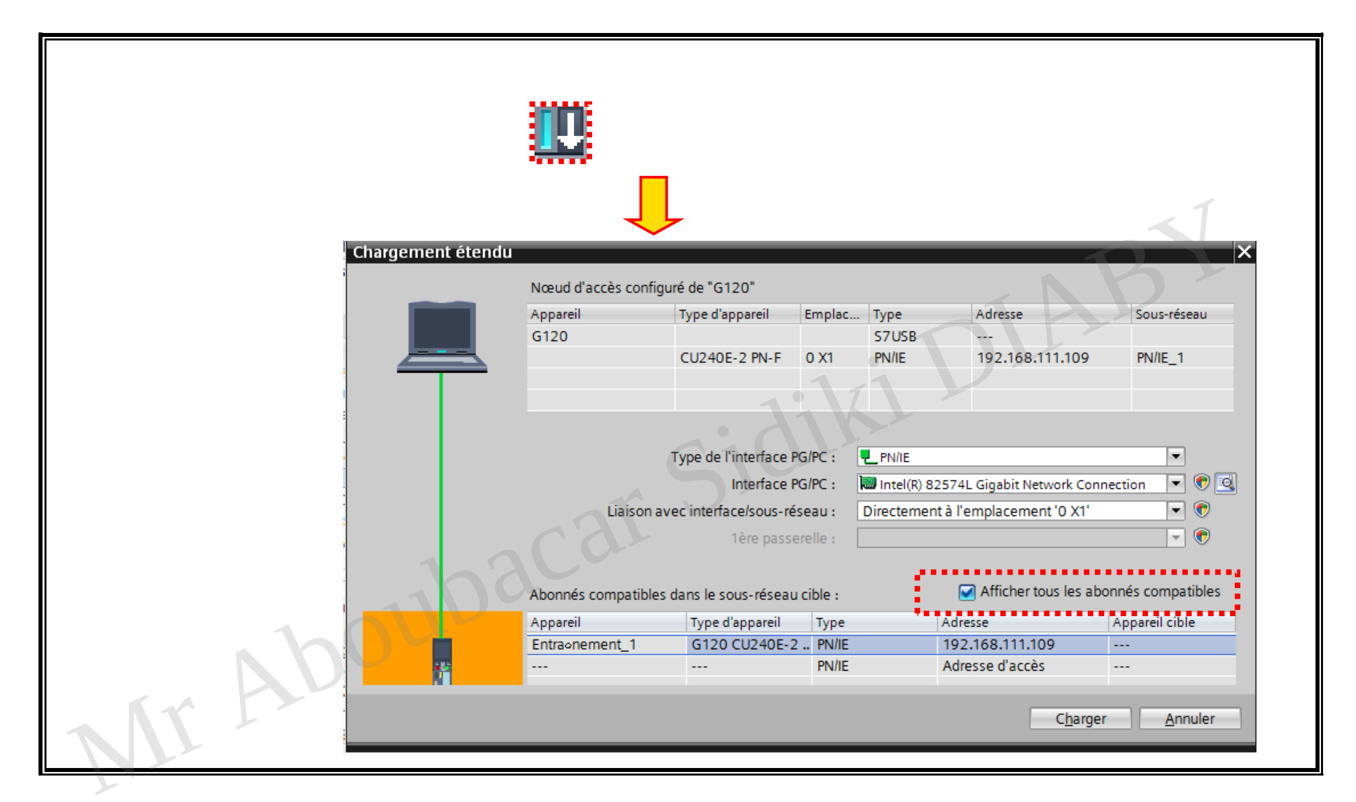
Réglage de l'entraînement:
Norme mot CEI/NEMA: [0] Moteur CEI (50 Hz, unités SI)
Tension de raccordement des variateurs: 230 V
Partie puissance Application: [0] Cycle de charge avec forte surcharge p. entraînements vectoriels

Moteur:
Type moteur Sélect: [1] Moteur asynchrone
Moteur Type de couplage: Triangle
Moteur Fonctionnement 87 Hz: Non
Tension assignée du moteur: 230 Veff
Courant assigné du moteur: 0,73 Aeff
Puissance assignée du moteur: 0,12 kW
Fréquence assignée du moteur: 50,00 Hz
Vitesse assignée du moteur: 1350,0 1/min
Mode de refroidissement du moteur: [0] Refroidt nat.

Paramètres importants:
Limite de courant: 1,10 Aeff
Vitesse de rotation minimale: 0,000 1/min
Vitesse rot. max: 1500,000 1/min
Générateur de rampe Temps de montée: 1,000 s
Générateur de rampe Temps de descente: 1,000 s

10. Terminer

16.5.5. Exercice 6: Charger le paramétrage dans le variateur



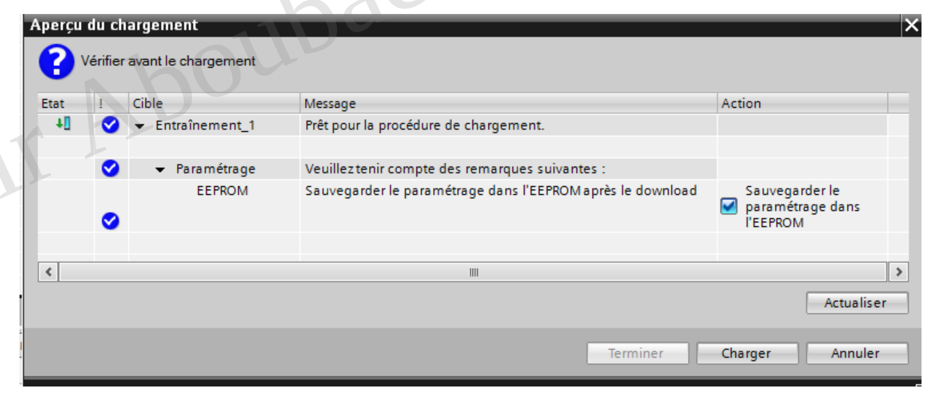
Enoncé

Suite au paramétrage de base réalisé Hors Ligne, vous allez à présent les charger dans le variateur.

Marche à suivre

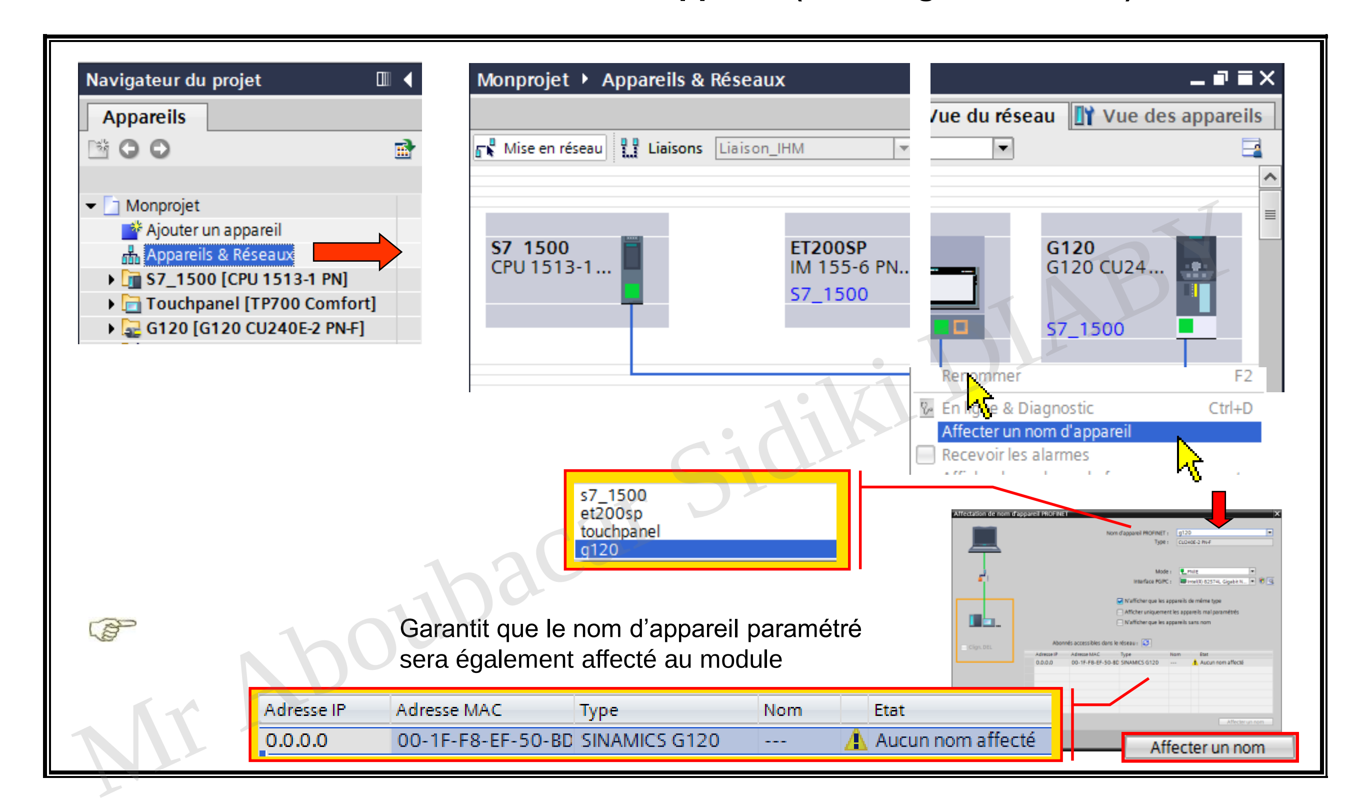
1. Sélectionnez le variateur dans le Navigateur du projet et lancez l'opération de chargement de la configuration matérielle et du paramétrage avec le bouton « Charger dans l'appareil ».
2. Un dialogue apparaît « Chargement étendu » ce dernier contient la liste des « Abonnés compatibles dans le sous-réseau cible ». Si le variateur ne s'y trouvait pas, veuillez activer l'option « Afficher tous les abonnées accessibles » (voir figure)
3. Choisissez votre variateur dans la liste des « Abonnés compatibles dans le sous-réseau cible » et démarrez par « Charger ».

   Pour ne pas perdre le paramétrage du variateur sur coupure de la tension d'alimentation, veuillez renseigner le dialogue comme indiqué ci dessous:



4. Chargez également la configuration matérielle de la station S7-1500

16.5.6. Exercice 7 : Affecter un nom à l'appareil (nommage du module)



Enoncé

Affectez au variateur le nom d'appareil PROFINET paramétré.

 Même si un nom s'affiche dans les « Abonnés accessibles », le nom d'appareil PROFINET affecté a été effacé par « Rétablir les réglages usine ».

Marche à suivre

Ouvrez « Appareils & Réseaux » dans le navigateur du projet, puis sélectionnez la fonction dans le menu contextuel (clic droit) de la liaison « PN/IE ».

Sélectionnez le nom d'appareil paramétré et l'appareil cible correspondant, puis affectez le nom.

LED du variateur :
→ BF – s'éteint alors qu'elle clignotait en rouge
→ RDY – s'allume en vert

16.5.7. Exercice 8 : commander le variateur via le tableau de commande



Enoncé

Comme le variateur est configuré et qu'une adresse IP lui a été affectée, il peut dès à présent être commandé via le tableau de commande.

Marche à suivre

1. Activer le tableau de commande

Ouvrez l'éditeur « Mise en service » et activez le « Tableau de commande ».

2. Tester le variateur via le tableau de commande



Activer la maîtrise de commande

Le variateur peut à présent être commandé manuellement dans la fenêtre « Forcer ».

3. Désactiver le tableau de commande

Désactiver — Désactiver la maîtrise de commande.

4. Fermer la « Mise en service »

Fermez l'éditeur « Mise en service »

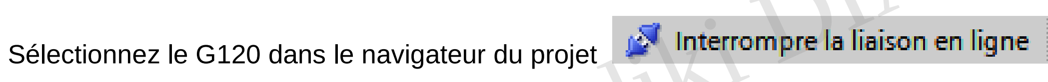


Si la maîtrise de commande n'a pas été désactivée, un message d'erreur s'affiche sur le variateur (généralement F1030 – Absence de signe de vie avec maîtrise de commande activée)



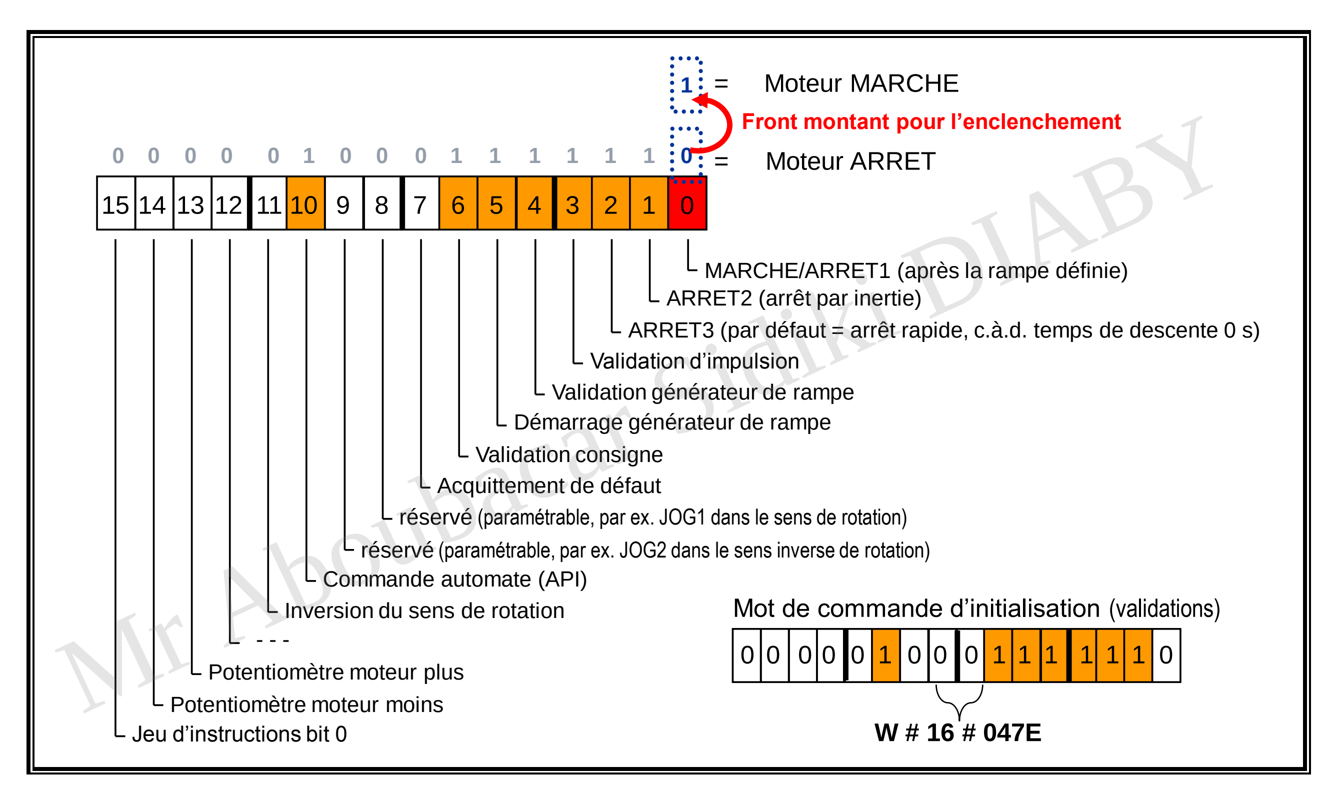
Bouton pour l'acquittement/effacement des défauts

5. Interrompre la liaison en ligne avec le G120

Sélectionnez le G120 dans le navigateur du projet 

.

16.6. Structure du mot de commande (CDS0 – Control DataSet pour commande via bus de terrain)



Le variateur peut être commandé de différentes manières. Le G120 peut ainsi être commandé via le bus de terrain ou via le bornier.
Le mode est commandé par le jeu de paramètres de commande correspondant (Control Data Set, CDS).

Réglage par défaut sur le G120 = deux CDS :
- Jeu de paramètres de commande CDS0 : commande du variateur via le bus de terrain
- Jeu de paramètres de commande CDS1 : commande du variateur via le bornier.

Mot de commande

Le mot de commande (bits 0-10) est conforme au profil standard PROFIdrive. Les bits 11-15 sont spécifiques au variateur.

 Pour des raisons de sécurité en cas de rupture de fil, la désactivation du variateur doit toujours s'effectuer lorsque le signal du bit d'arrêt correspondant est à 0.

MARCHE/ARRET1

Pour l'enclenchement (activation) du variateur, un front montant est en principe nécessaire sur ce bit et les bits ARRET2 et ARRET3 doivent être à l'état 1.
Lors du déclenchement (désactivation) via ce bit, le moteur s'arrête selon la rampe de décélération du générateur de rampe (paramètre 1121), puis le variateur est coupé.

ARRET2

Le variateur est coupé immédiatement et le moteur s'arrête en roue libre. Pour réenclencher le variateur, ce bit doit à nouveau être mis à 1 et un front montant est nécessaire à MARCHE/ARRET1.

ARRET3

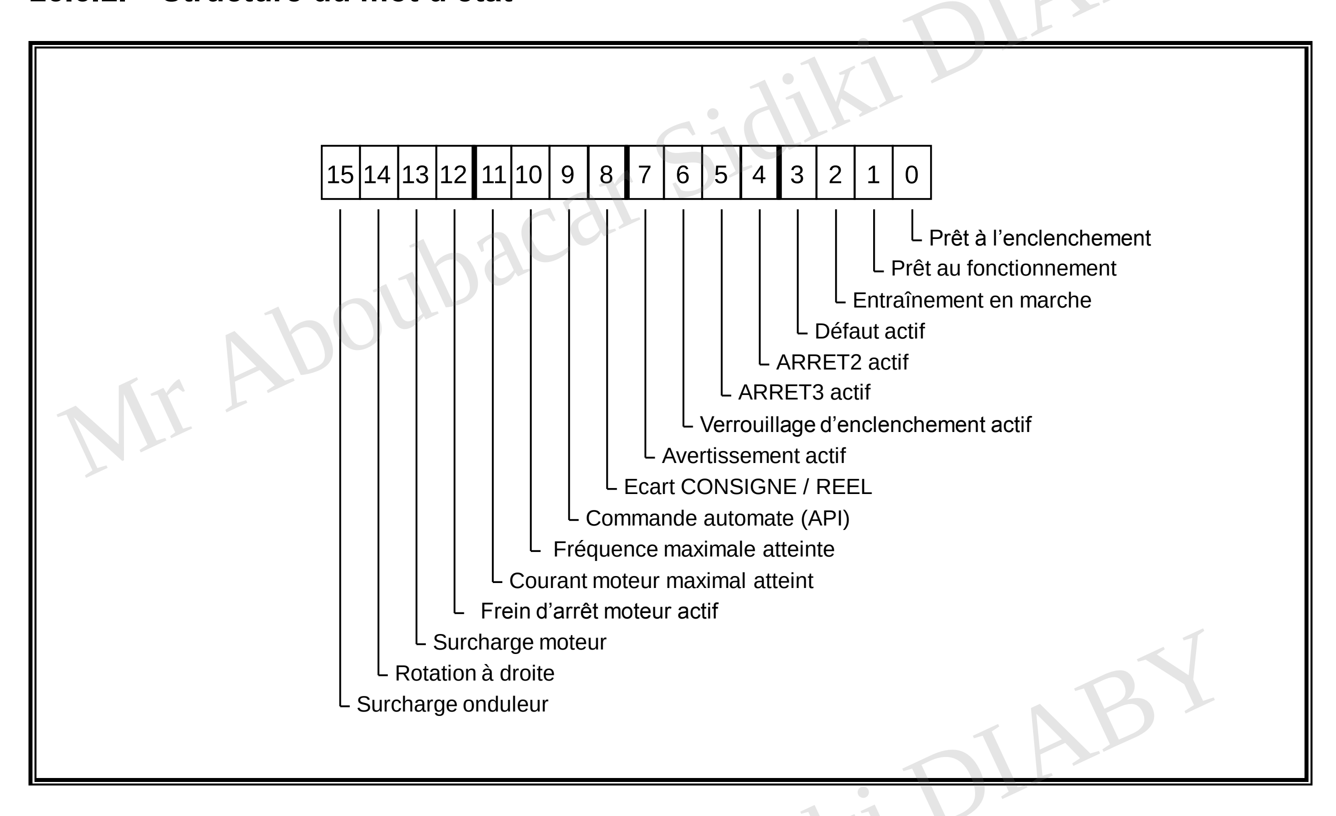
Le moteur s'arrête selon la rampe de décélération ARRET3 et le variateur reste enclenché. Cette fonction est souvent utilisée comme Arrêt d'urgence.

→ Réglage par défaut pour ARRET3 = 0 s (difficilement réalisable en pratique pour les gros moteurs)

Initialisation du variateur

Avant que le variateur puisse être enclenché, les ordres de déblocage doivent avoir été transmis. Ces ordres de déblocage peuvent être programmés lors du démarrage de la CPU (par ex. OB100).

16.6.1. Structure du mot d'état



Mot d'état

Le mot d'état (bits 0-10) est conforme au profil standard PROFIdrive.

Les bits 11-15 sont spécifiques au variateur.

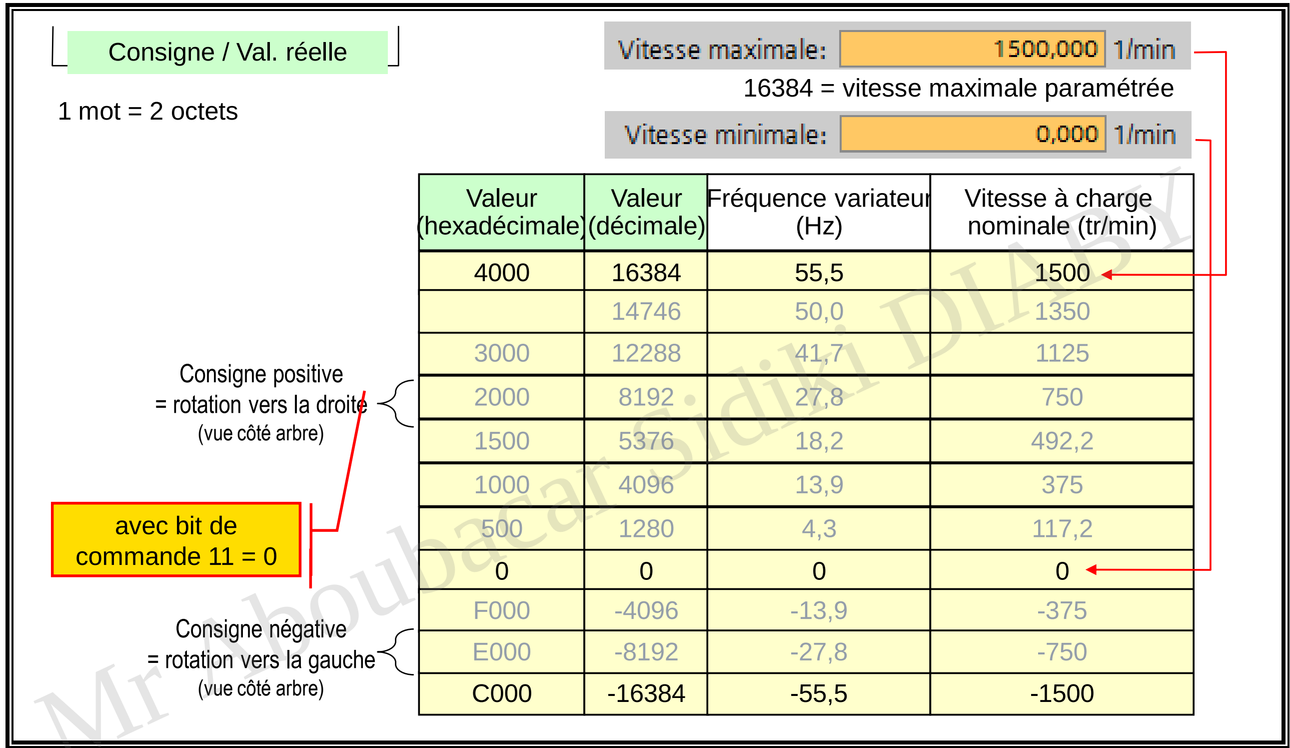
Blocage du ré-enclenchement

Cet état se présente après l'élimination d'un défaut et l'acquittement d'une défaillance du variateur. Le ré-enclenchement du variateur n'est possible que via MARCHE/ARRET1 '0' →' 1'.

Avertissement, alarme

Alarmes (bit 13, bit 15) et avertissements (bit 11)

16.6.2. Consigne/mesure → valeurs de vitesse



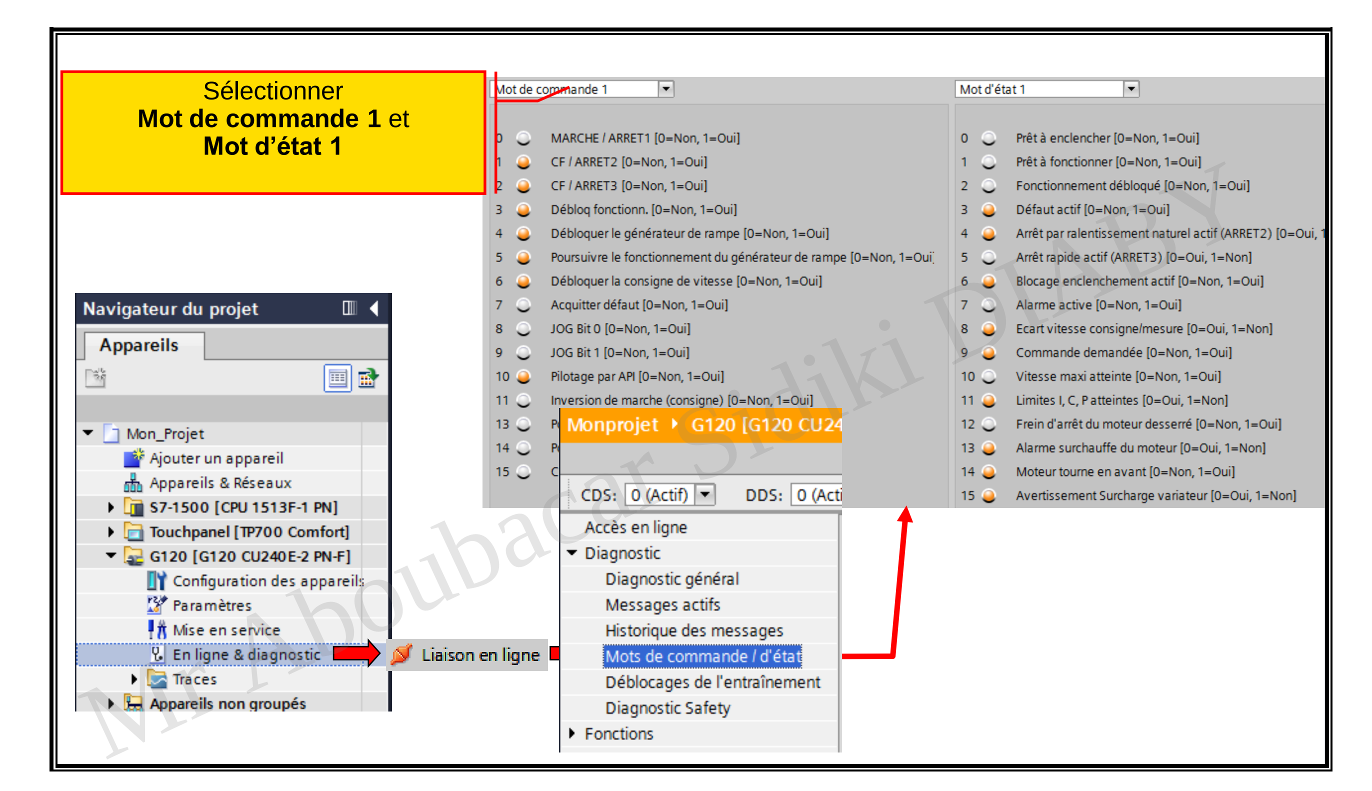
Valeur (hexadécimale)	Valeur (décimale)	Fréquence variateur (Hz)	Vitesse à charge nominale (tr/min)
4000	16384	55,5	1500
	14746	50,0	1350
3000	12288	41,7	1125
2000	8192	27,8	750
1500	5376	18,2	492,2
1000	4096	13,9	375
500	1280	4,3	117,2
0	0	0	0
F000	-4096	-13,9	-375
E000	-8192	-27,8	-750
C000	-16384	-55,5	-1500

Mise à l'échelle des valeurs de vitesse

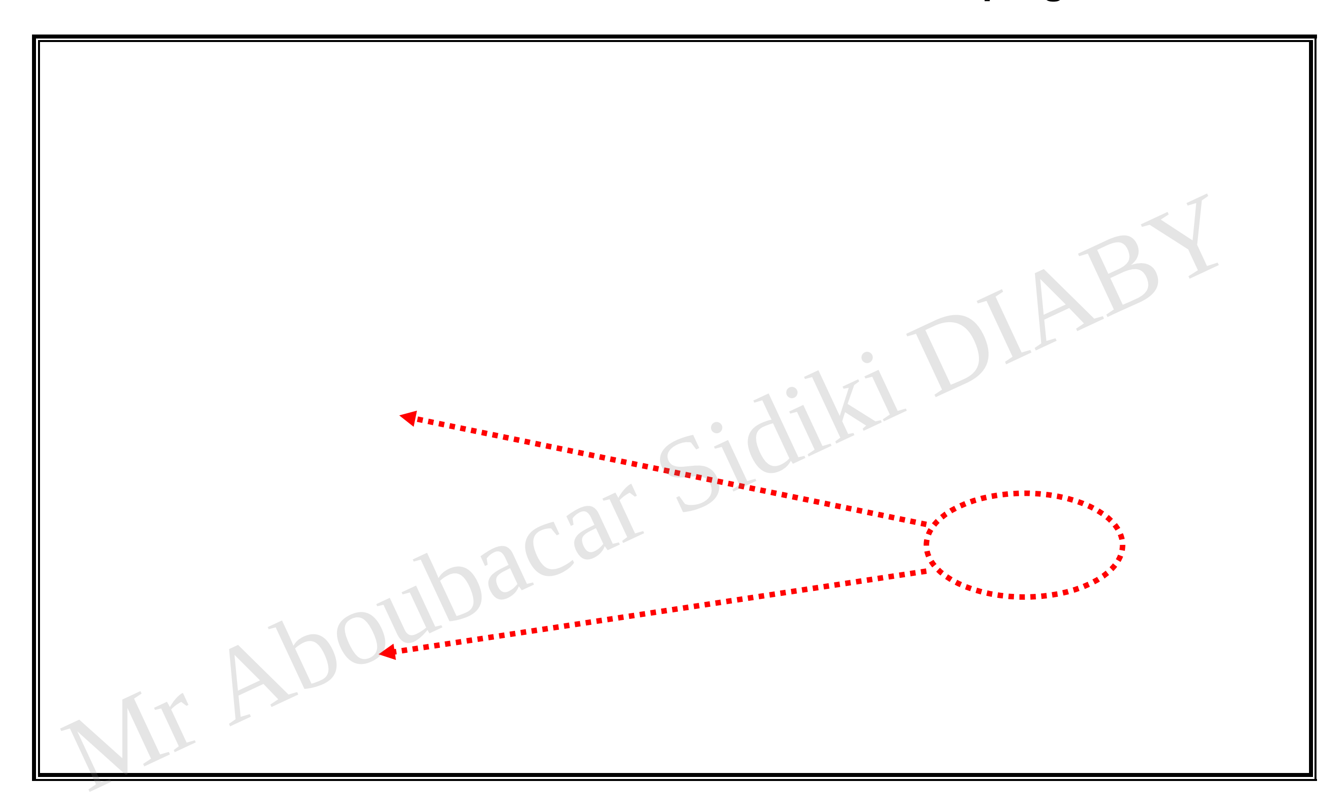
Via le paramètre p2000, une vitesse de référence est définie pour toutes les valeurs de consigne et de mesure.

La plage de -100% à 100% de la consigne et mesure, est mise à l'échelle dans l'intervalle 16384 ($CH00_{HEX}$) à +16384 (4000_{HEX}).

16.6.3. Visualiser en ligne les mots de commande et d'état



16.7. Exercice 9 : Mise en service de l’extension de programme



Enoncé

Imaginons que le G120 serve de dispositif d’entraînement du convoyeur lorsque des pièces sont transportées automatiquement à partir des postes 1 ou 2 en traversant la barrière lumineuse.

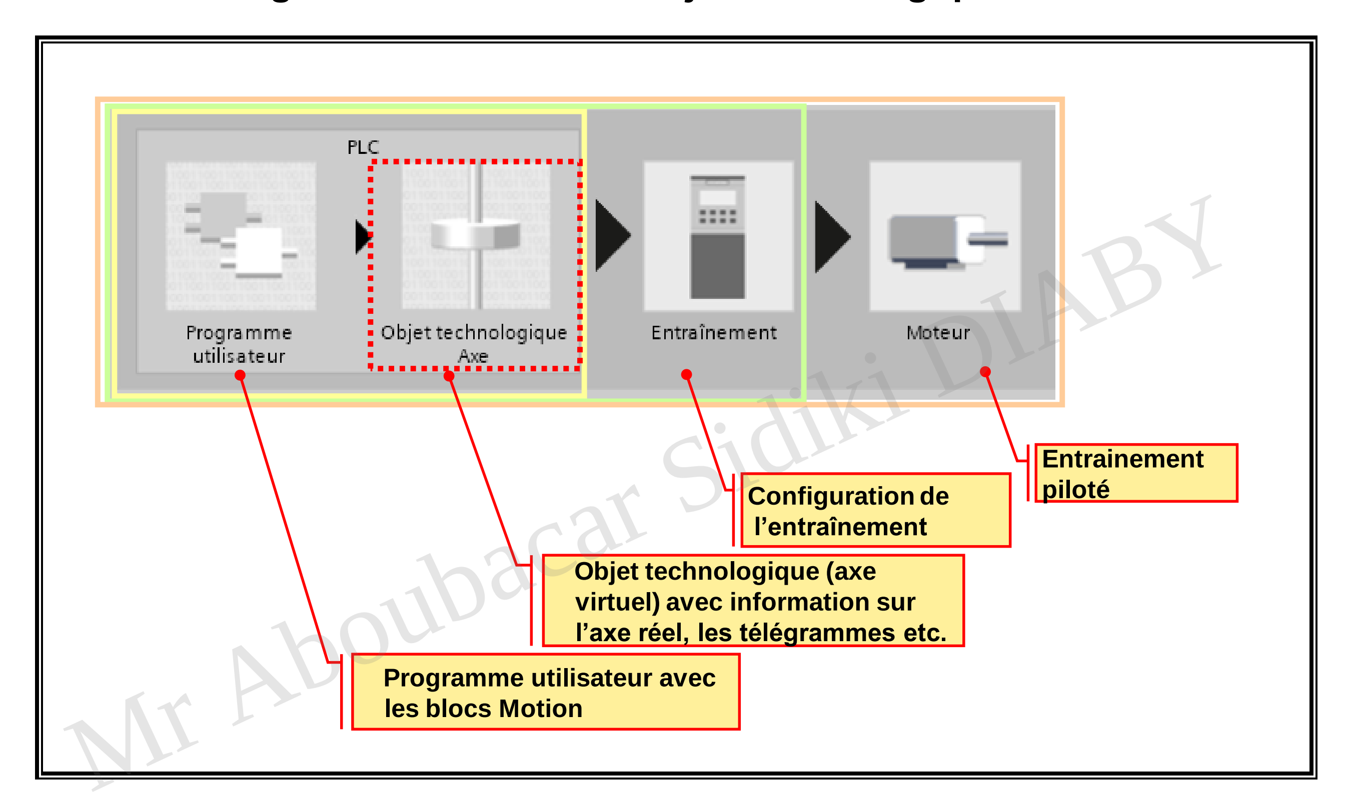
Dans ce cas le G120 avec « Q_En_Service » (Q0.1) sera piloté en parallèle au moteur de la bande transporteuse. La vitesse de rotation du G120 est réglable avec le curseur du potentiomètre à droite.

La fonction est réalisée dans le bloc « FC_Entraînement » (FC120) et doit être copié à partir de la bibliothèque puis mis en service.

Marche à suivre

1. Ouvrez la bibliothèque « Bib_TIA_PRO1 »
2. Copiez les éléments suivants par glisser-déposer « FC_Entrainement » (FC120) et « Var_AP_Entraînement » (Table de variables) dans votre projet
3. programmez l’appel du « FC_Entrainement » (FC120) dans OB1
4. Chargez tous les blocs dans la CPU et vérifiez qu’avec « Q_En_Service » (Q0.1) le G120 soit piloté en parallèle au moteur de la bande.
5. Vérifiez si le curseur de droite vous permet de changer la vitesse de rotation du G120.
7. Sauvegardez votre projet.

16.8. Pilotage d'un axe avec un Objet Technologique

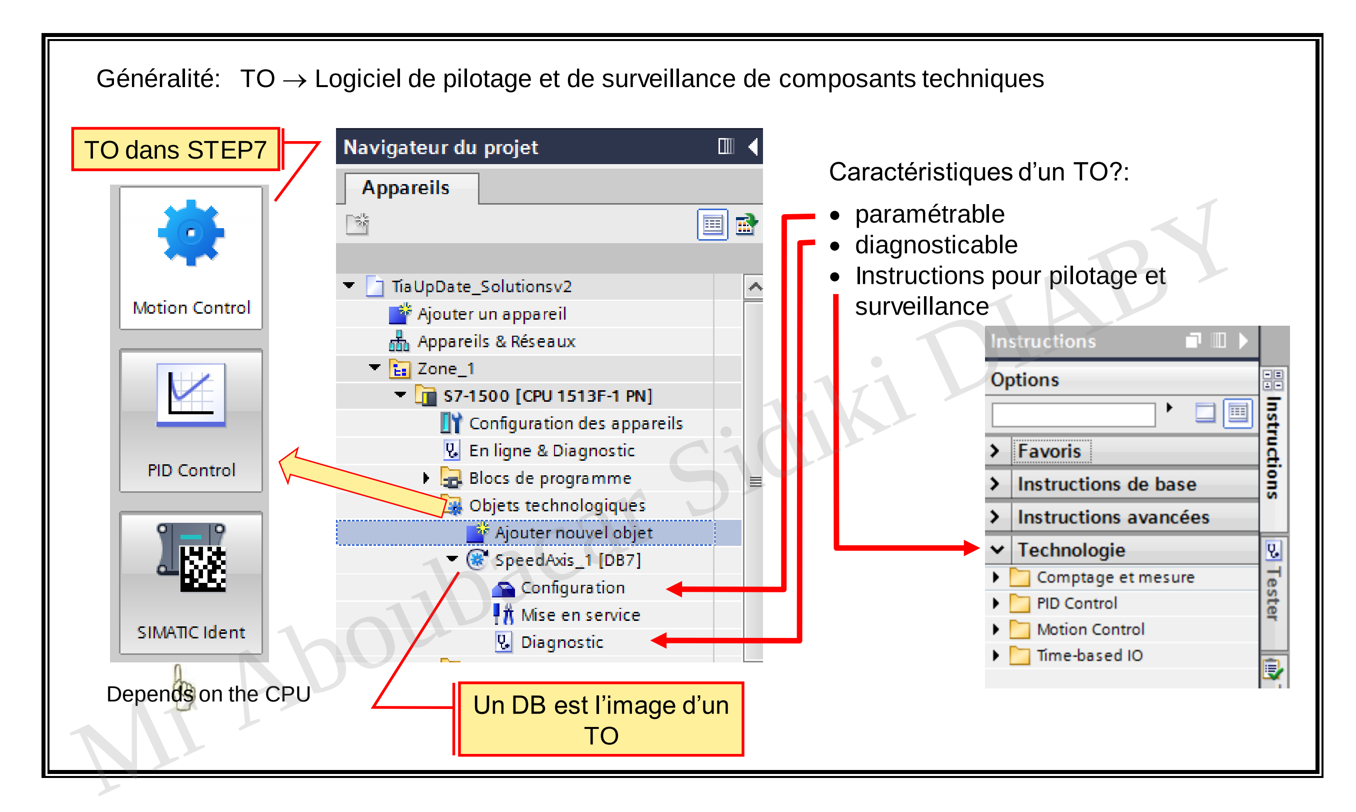


Principe du pilotage d'un axe

Pour la commande des axes vous disposez d'un modèle décliné en quatre sections qui vous aidera avec des assistants et des vues de diagnostic lors de la mise en service et du diagnostic des axes.

- Programme utilisateur
  Le programme utilisateur se base sur des instructions normalisées « Motion Control » pour le pilotage de l'objet technologique « Axe » donc de l'axe ou de l'entraînement.
- Objet technologique « Axe »
  L'objet technologique « Axe » représente un axe virtuel dans le contrôleur. A l'aide d'un l'assistant vous pouvez facilement renseigner les caractéristiques de l'axe réel, le télégramme utilisé, les transmissions éventuelles et les paramètres de l'axe. En fait, l'objet technologique est un bloc de données avec une structure exactement prédéfinie dans laquelle se trouvent les paramètres d'entrée que vous avez saisis. L'indication de ce DB sera nécessaire pour la programmation des instructions Motion Control du programme utilisateur. Au final, il est ainsi possible de programmer ces instructions de manière simple et compréhensible. La transposition en valeurs normées de ces instructions avec, par exemple, la génération du mot de commande ou la conversion de la vitesse en tenant compte du rapport de réduction est assurée par l'objet technologique. L'utilisateur n'a pas besoin de connaître la structure du mot de commande ou encore la manière de convertir la vitesse de rotation.
- Entraînement
  Pour convenir, tous les entraînements ou moteurs doivent disposer soit d'un pilotage via une interface réseau (PN, DP) au profil PROFIdrive, soit d'une interface analogique (consigne, valeur réelle) ou d'une interface avec signaux impulsions/direction.

16.8.1. Les Objets Technologiques (OT) dans SIMATIC S7



Généralités d'un Objet Technologique (OT)

Les Objets Technologiques (Abréviation: OT) fournissent la couche logicielle pour piloter et surveiller des composants techniques.

Ils nécessitent, en fonction du OT, différentes entrés/sorties pour le pilotage du processus et la surveillance des actionneurs ou codeurs.

Les TO SIMATIC

En fonction de la CPU, différent OT sont proposés:

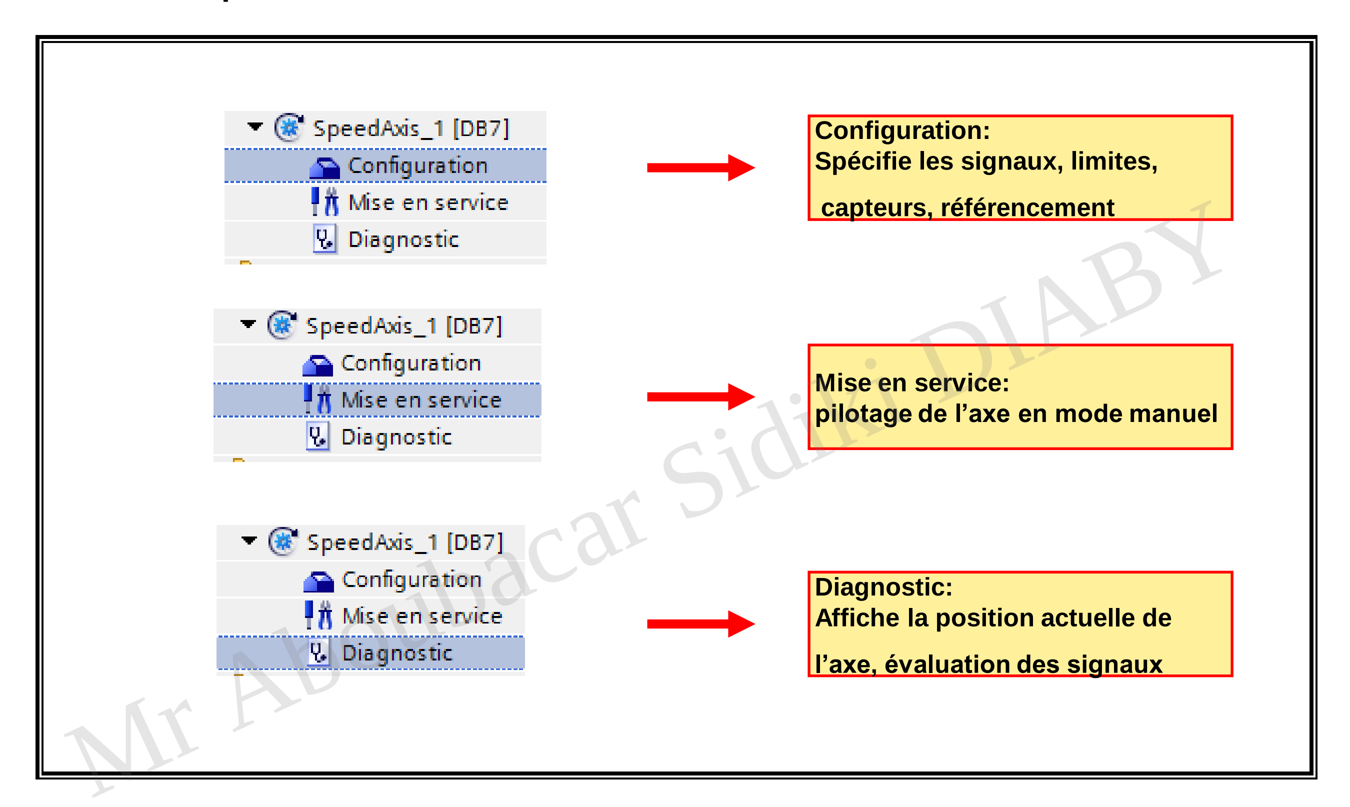
1 Motion Control → commande d'axe

2 PID (Control) → régulateur PID

3 SIMATIC Ident → Paramétrage d'un système SIMATIC Ident

4 Comptage et mesure → compteurs rapides

16.8.2. Propriétés des OT



Après la création de l'objet technologique « Axe », vous disposez de trois choix pour le traitement:

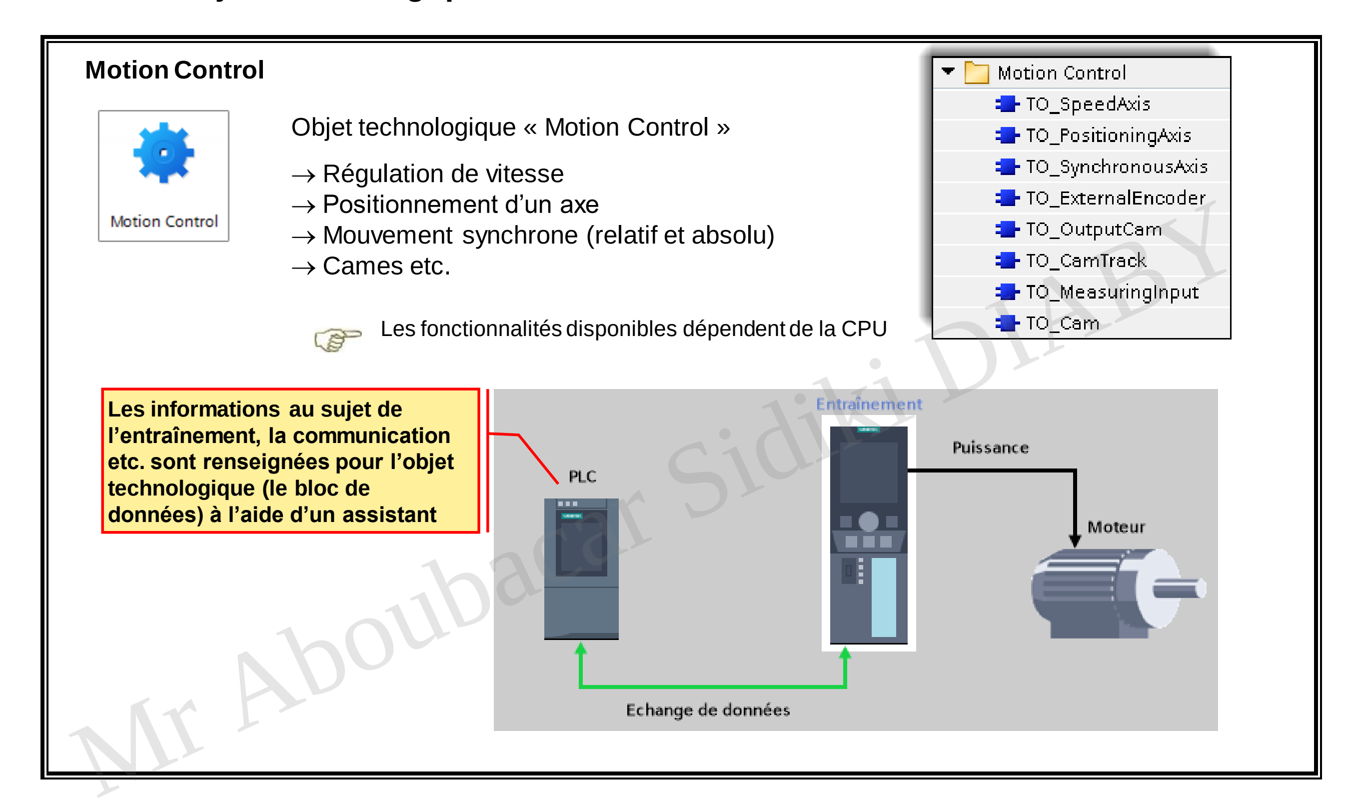
- **Configuration**
  - Sélection de la connexion et de l'entraînement.
  - Propriétés de la mécanique et du rapport de réduction, le télégramme,
  - Les paramètres dynamiques,

  La configuration est sauvegardée dans le bloc de données de l'objet technologique
- **Mise en service**

  Avec cet outil le fonctionnement de l'axe est testé sans programme utilisateur. Lorsque vous lancer l'outil vous ouvrez le panneau de commande. Les commandes suivantes sont disponibles au niveau du panneau:
  - Valider ou désactiver l'axe
  - Démarrer avec une certaine vitesse
  - Acquitter les erreurs
- **Diagnostic**

  Avec l'outil vous contrôlez l'état et les informations d'erreurs de l'axe et de l'entraînement.

16.8.3. Objet Technologique « Motion Control »



Dans l'automatisation, Motion Control fait référence au contrôle des mouvements.

Il s'agit de régulation de position, de vitesse, d'accélération ou d'une combinaison de ces éléments.

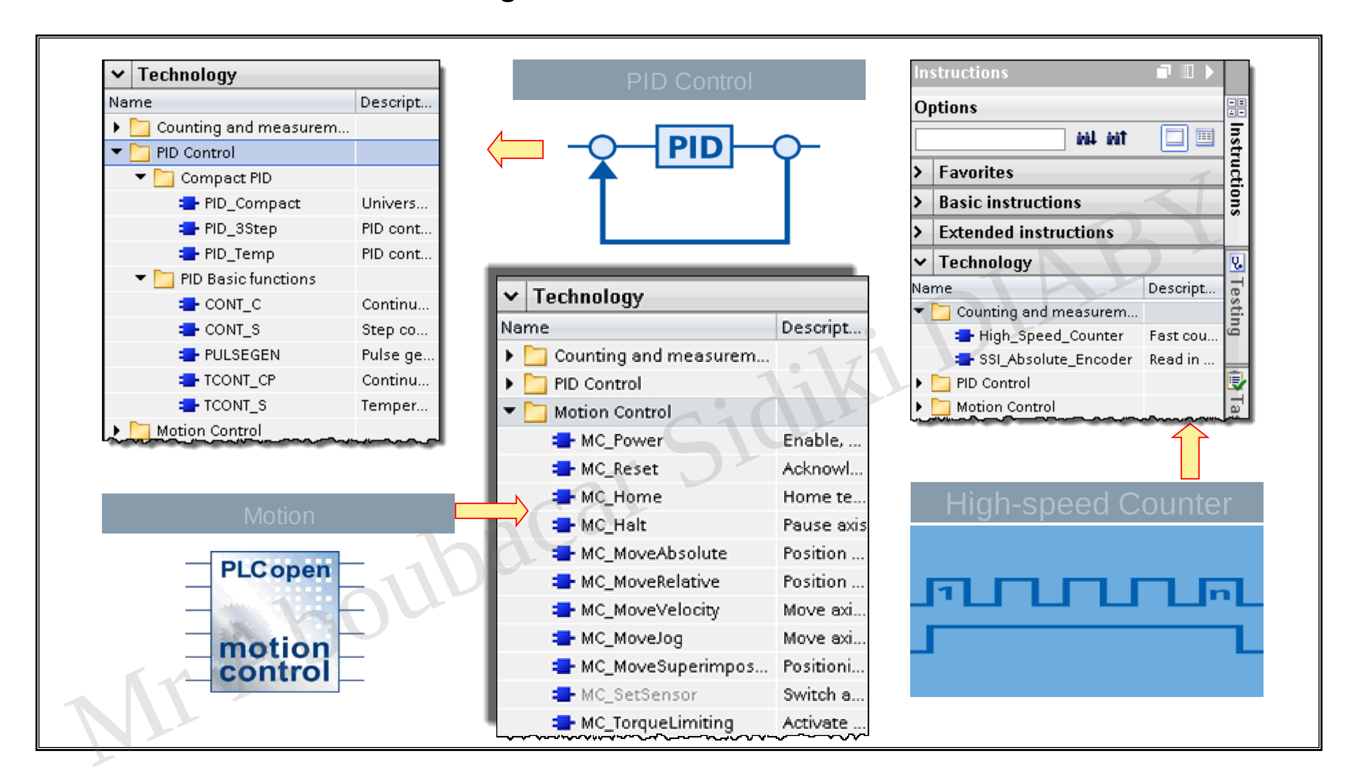
Un moteur électrique sert d'actionneur et de codeur pour la régulation.

Applications courantes

- Régulation de vitesse
- Régulation de position Point à Point
- Régulation de synchronisme de vitesse
- Pilotage de cames

Un axe virtuel est créé (bloc de données) dans lequel les propriétés de l'entraînement sont sauvegardées à l'aide d'un assistant. A l'aide de ce DB, l'entraînement peut être piloté le plus simplement.

16.8.4. Fonctions « Technologie »



Compteurs rapides

Vous disposez de compteurs rapides jusqu’à 100 kHz (dépendant de la CPU) pour l’acquisition des signaux de codeurs incrémentaux, de fréquences ou pour le comptage des évènements d’un processus rapide.

Motion

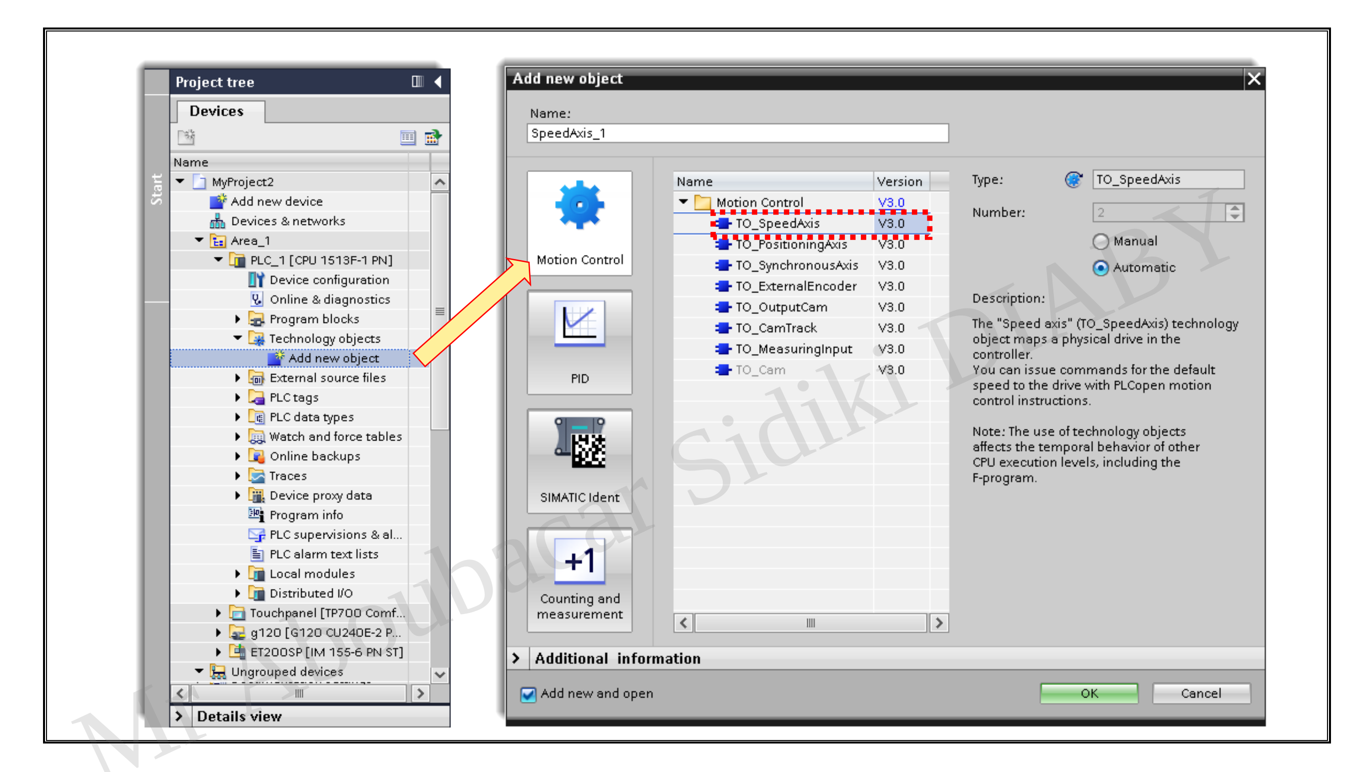
Pour le pilotage de la vitesse, position ou du rapport cyclique vous disposez de sorties PWM (modulation de la largeur d’impulsions). Des exemples d’applications sont, par exemple, le pilotage d’une vitesse de rotation d’un moteur, la position d’un moteur pas à pas ou d’un servomoteur.

Pour la régulation de vitesse et de position vous disposez de sorties PTO 100 kHz (sortie train d’impulsions). Des exemples d’application sont la régulation de vitesse et de position pour moteurs pas à pas et servomoteur.

PID

Pour des tâches de régulation simples, vous disposez de régulateurs PID avec optimisation des paramètres intégrée.

16.9. Exercice 10: Objet Technologique « SpeedAxis »



Enoncé

Il s'agit d'insérer un nouvel objet technologique.

Marche à suivre:

1. Dans le répertoire des Objets Technologiques, insérez un nouvel objet.
2. Sélectionnez « Motion Control » > « TO_SpeedAxis »
3. Donnez le nom SpeedAxis_G120 à cet objet (DB)

16.9.1. Exercice 11: Paramétrage de l’Objet Technologique



Enoncé

Paramétrez l’objet technologique avec l’assistant:

Entraînement:

Echange de données:

Echange de données

Télégramme d'entraînement : Télégramme 1

Configuration de l'appareil

☑ Application automatique des valeurs de paramètre de l'entraînement dans l'appareil

Mécanique:

Mécanique d'entraînement

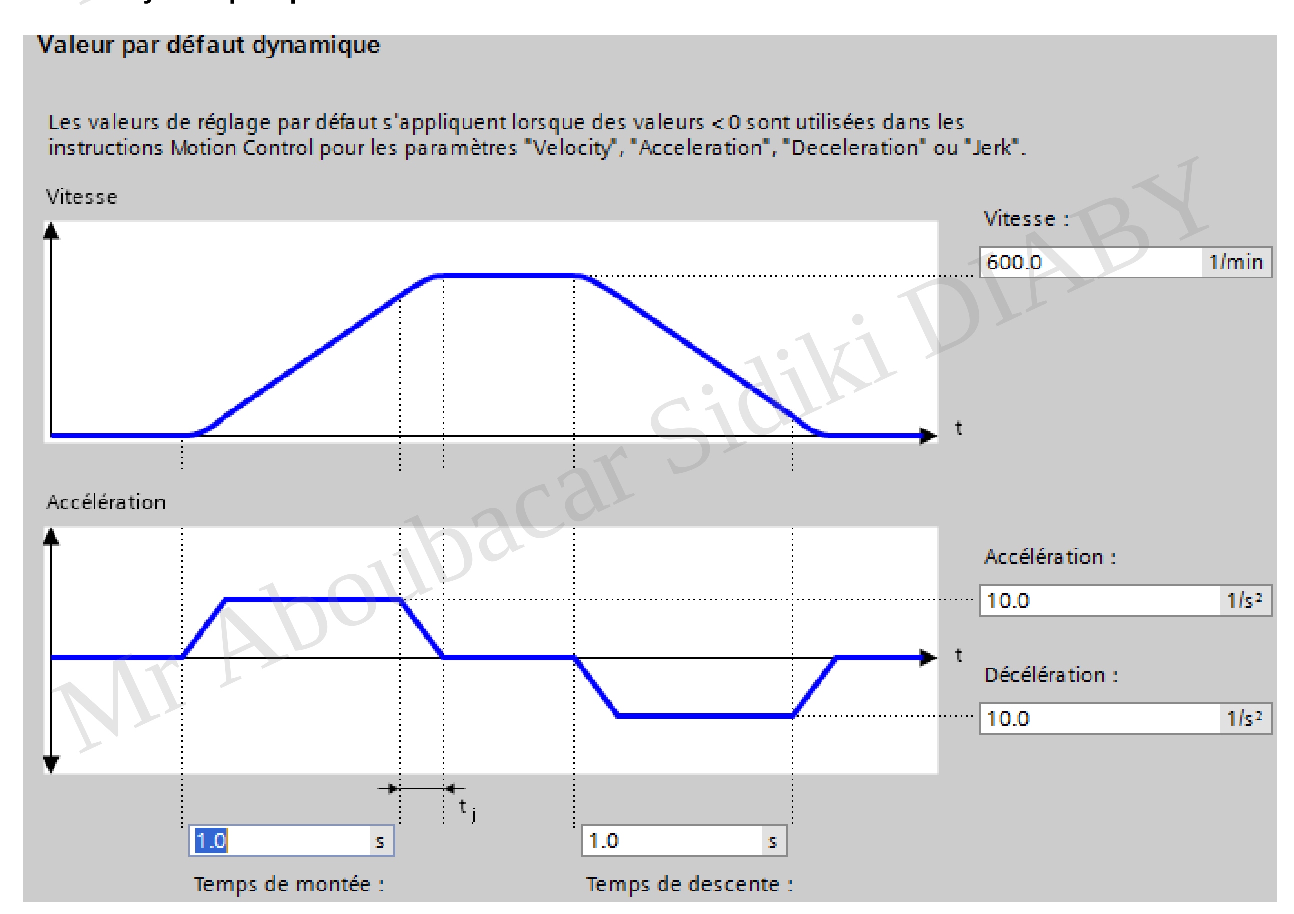
☐ Inverser le sens de l'entraînement

Réducteur principal

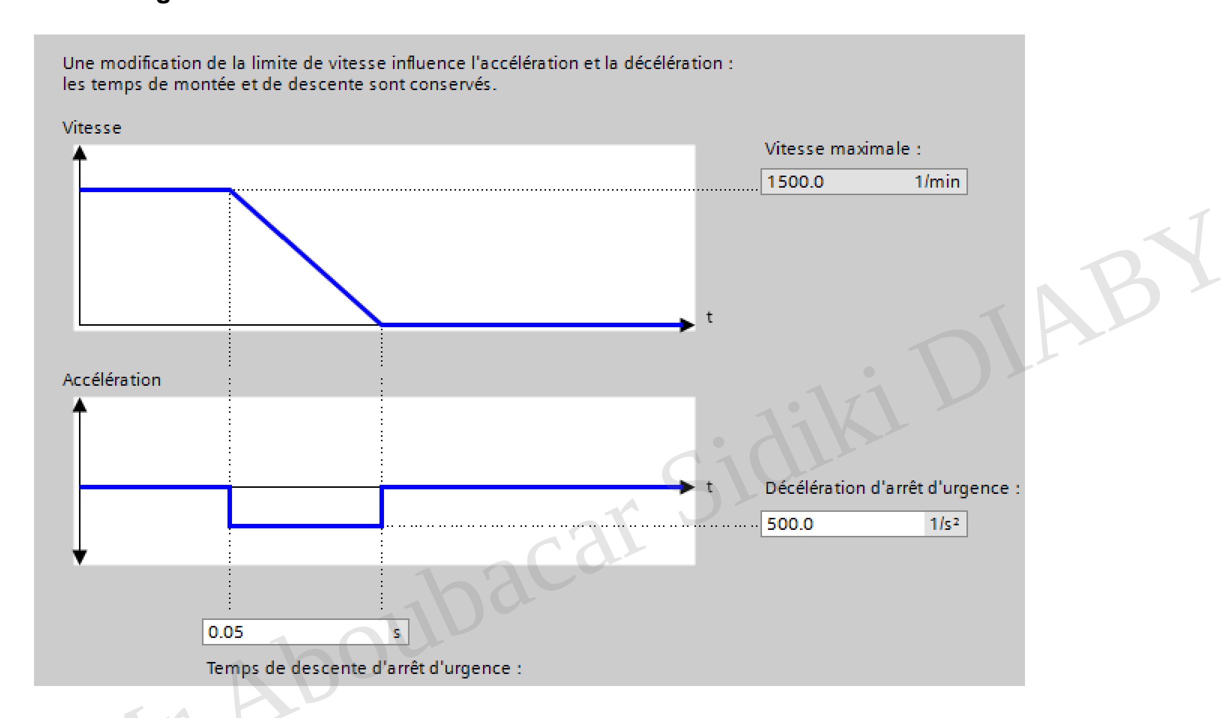
Nombre de tours du moteur : 1

Nombre de tours de charge : 1

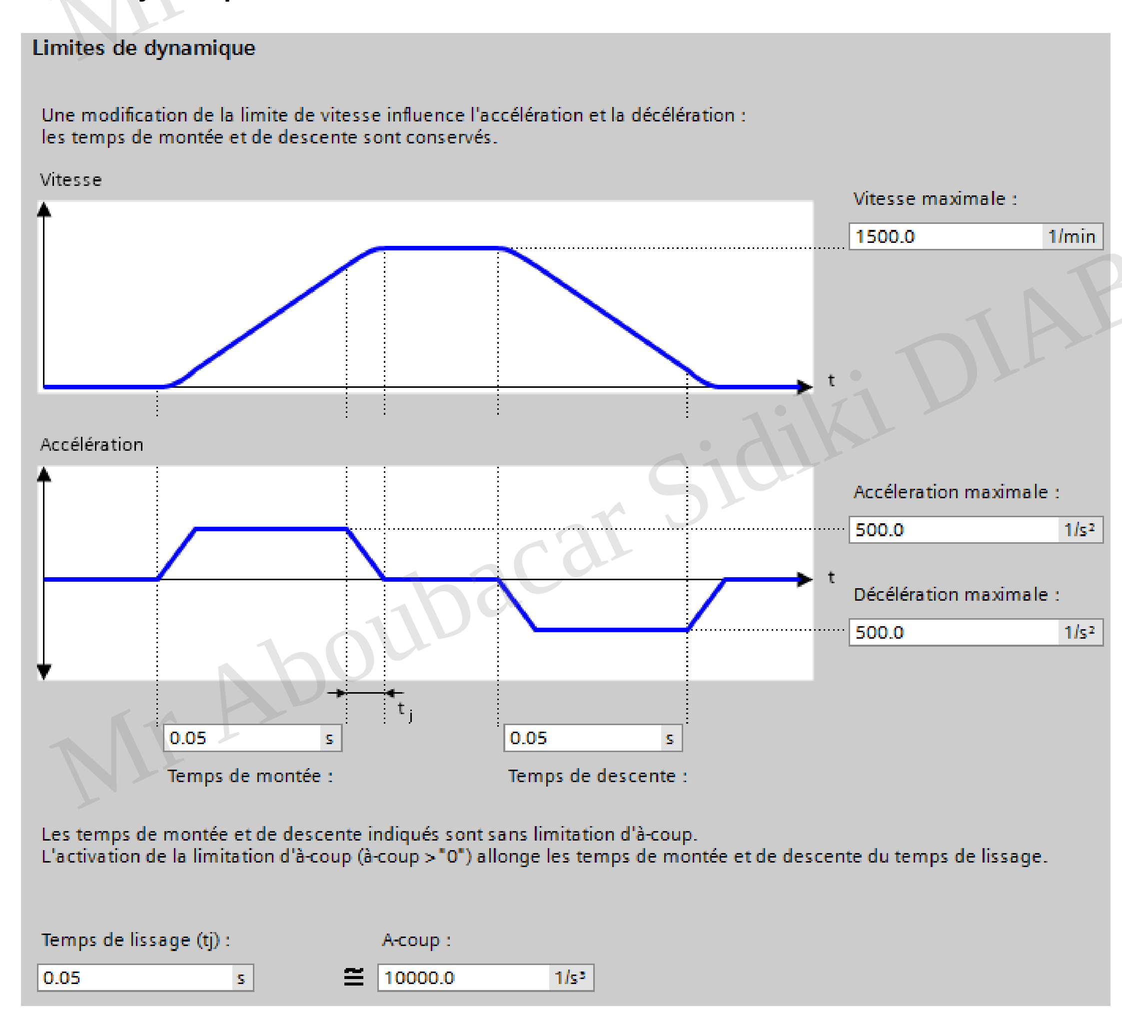
Valeurs dynamiques par défaut:



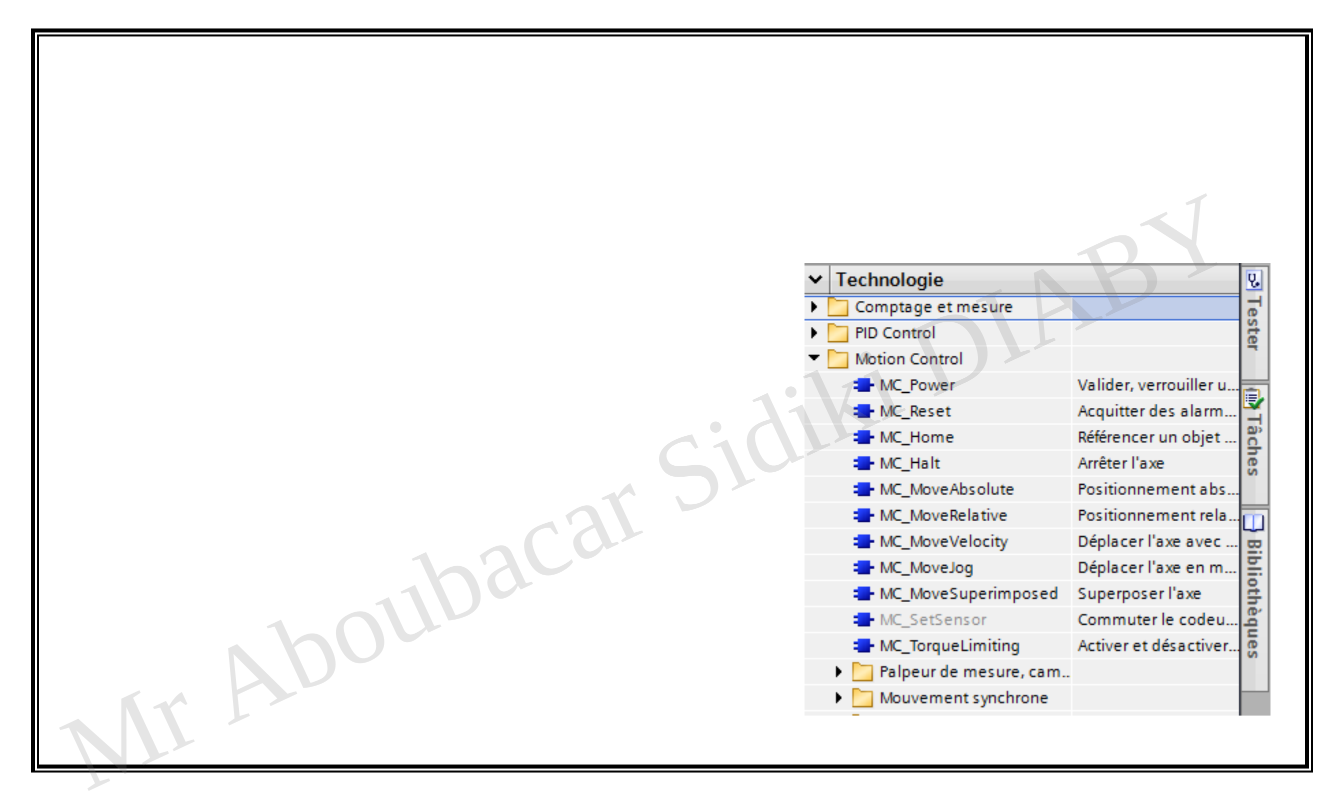
Arrêt d'urgence:



Limites dynamiques:



16.9.2. Exercice 12: pilotage de l'entraînement



Enoncé

L'entraînement peut maintenant être traité à l'aide des fonctions « Technologie ».

Marche à suivre:

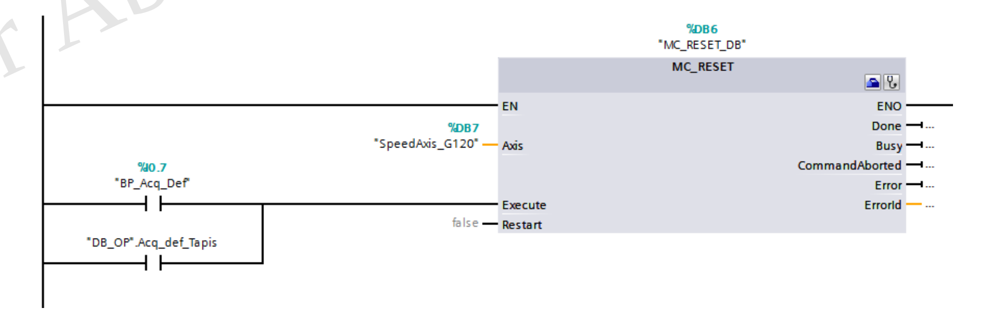
1. Modifiez le réseau 2 du FC_Entraînement de telle sorte que les valeurs réelles de la vitesse soient transférées dans la variable temporaire vitesse au format LReal.

2. Effacez le réseau 3

3. Utilisez la fonction « MC_Power » du Task Card > Technologie > Motion Control pour démarrer le moteur lorsque Q_En_service = True.
   La fonction nécessite un DB d'instance à choisir librement.
   Pour le paramètre « Axis » renseignez le avec l'objet technologique de telle sorte que le moteur correspondant soit adressé.

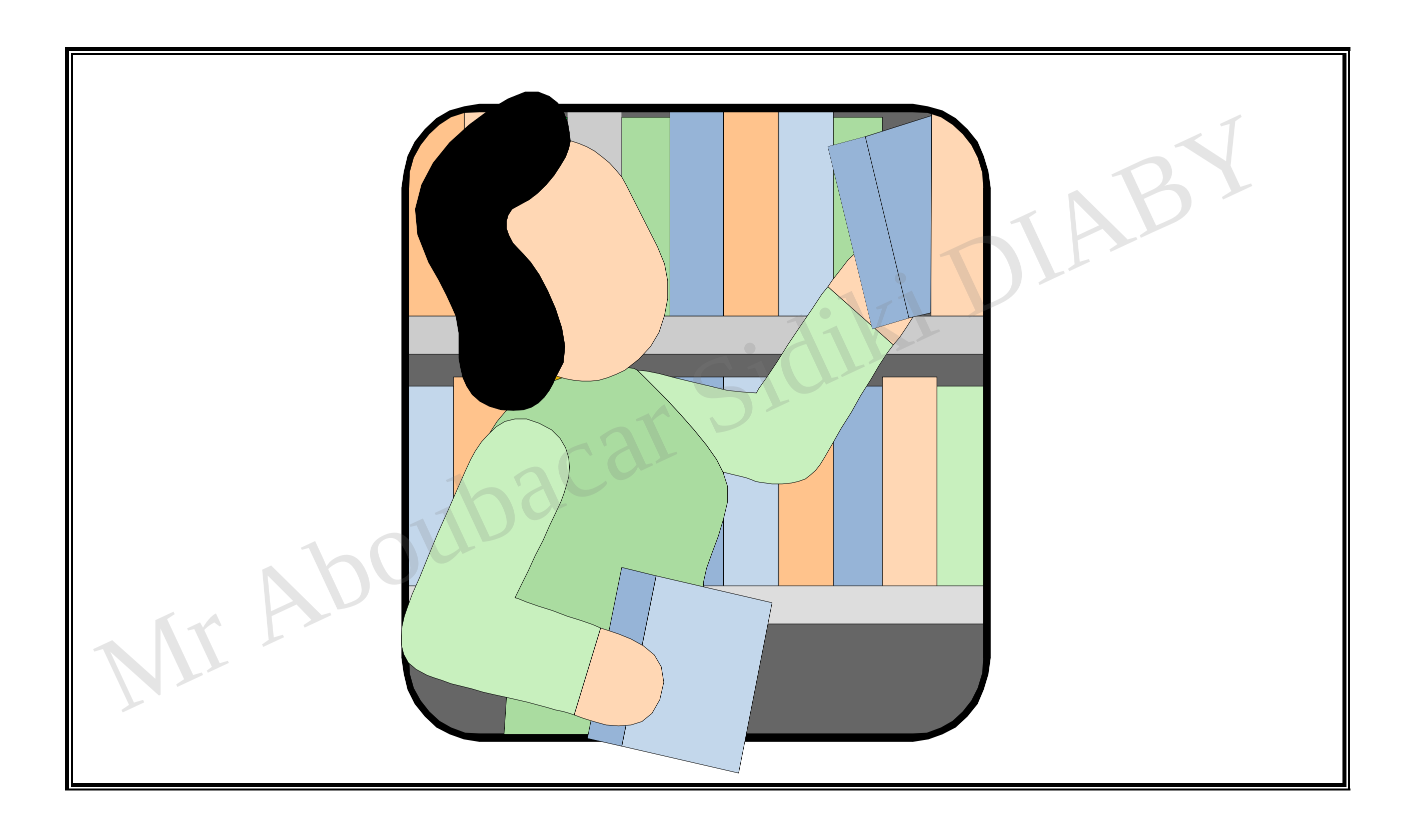
4. Utilisez la fonction « MC_MOVEJOG » pour pilotez le moteur avec la consigne et la direction.

5. Utilisez la fonction « MC_RESET » pour acquitter les erreurs en cours.



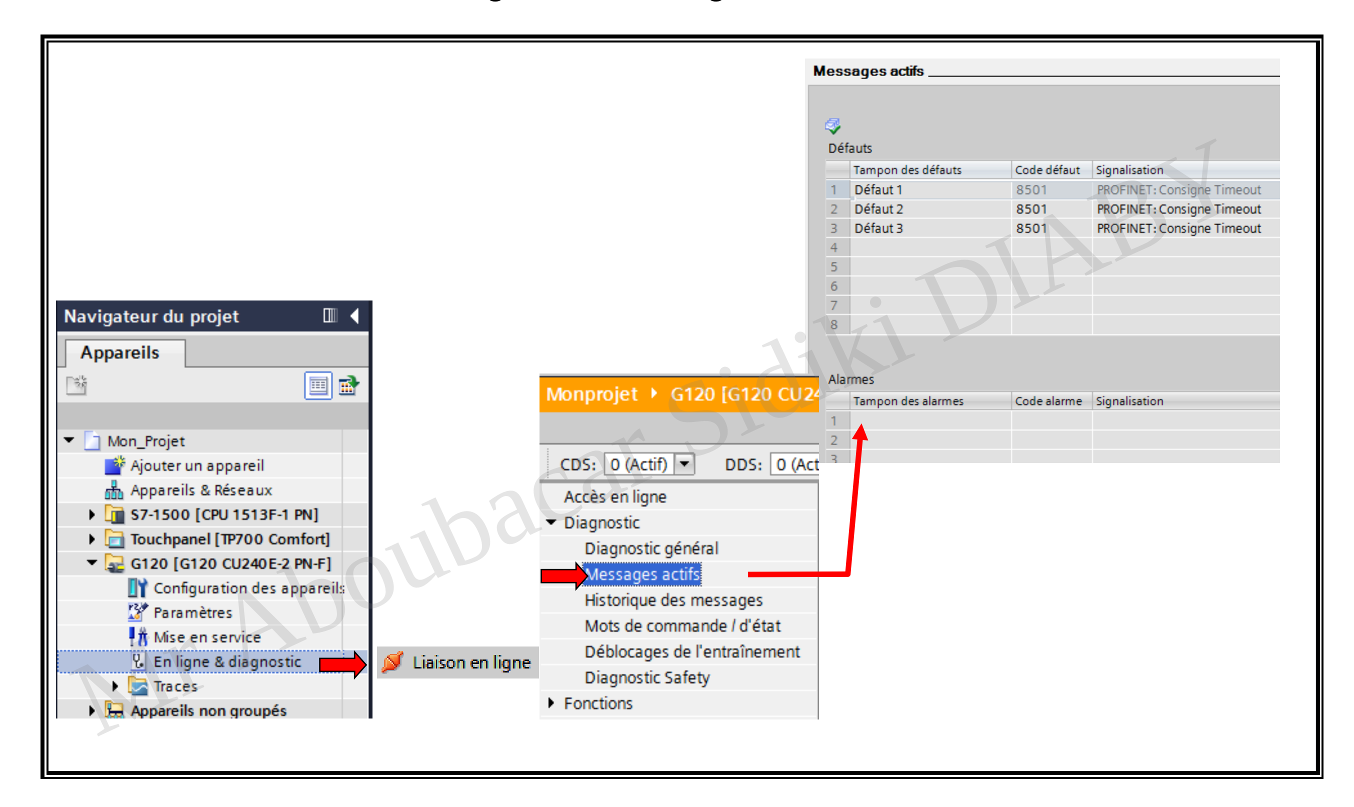
6. Sauvegardez le programme, chargez le dans la CPU et testez la fonction.

16.10. Plus d'informations

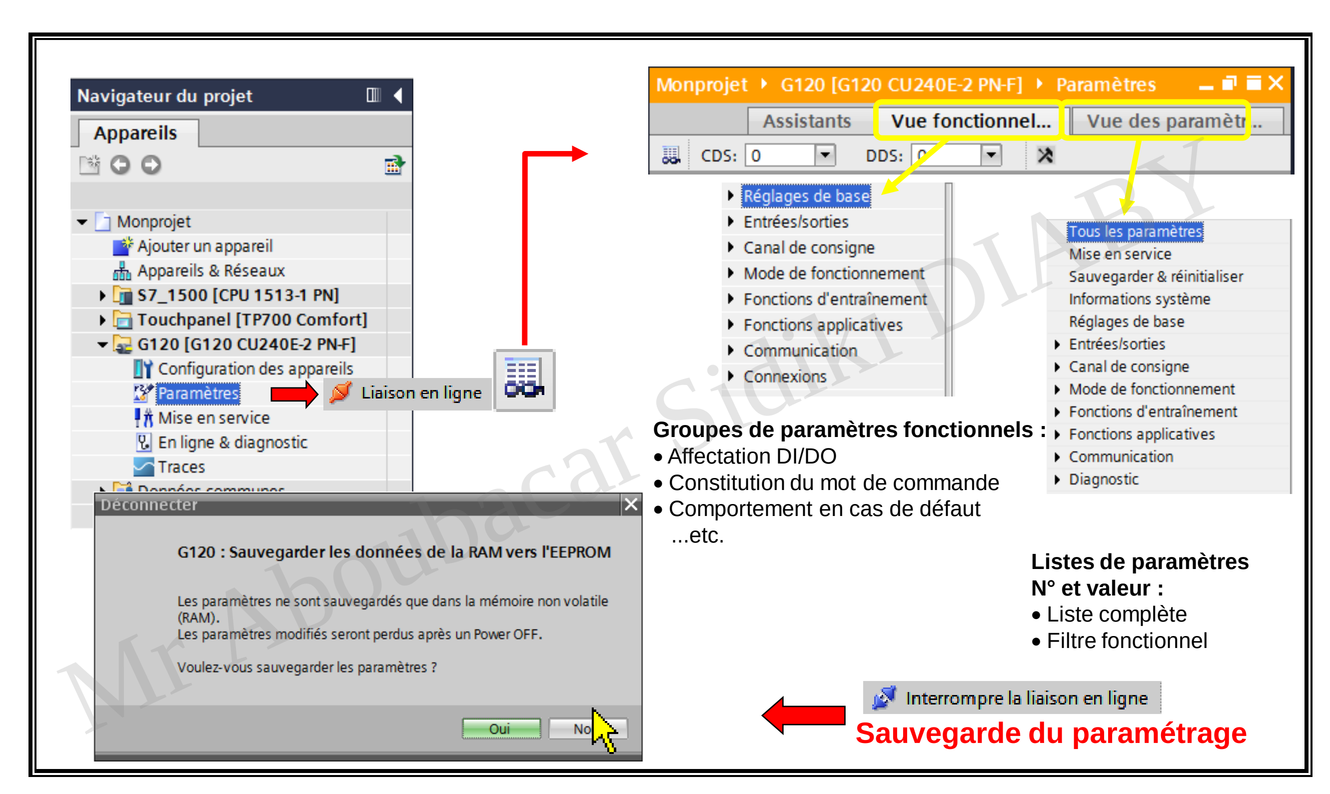




16.10.1. Visualiser les messages actifs en ligne



16.10.2. Modifier les paramètres dans le variateur



Les paramètres peuvent être modifiés via l'éditeur « Paramètres ».

Appel de l'éditeur « Paramètres » à partir de l'appareil du navigateur du projet

 Cette fonction doit être activée afin de pouvoir visualiser également les paramètres en ligne. Ces paramètres sont reconnaissables à leur affichage sur fond orange.

Lors de l'interruption de la liaison en ligne, le système demande si une copie vers l'EEPROM (sauvegarde des modifications de paramètres en cas de coupure de tension) doit être réalisée.

☞ Si la modification n'est réalisée qu'à des fins de test, vous pouvez répondre NON. Après coupure puis remise sous tension, le variateur travaillera alors à nouveau avec les valeurs précédentes.

Appel de l'éditeur « Paramètres » à partir de « Accès en ligne »

La modification de ces paramètres est directement enregistrée dans l'EEPROM.

16.10.3. Rétablir les réglages usine pour les paramètres du variateur



L'opération de réinitialisation (affichage « BUSY ») peut durer quelques secondes.